N73-27783

NASA CONTRACTOR REPORT

NASA CR-2268

NASA CR-2268

PERFORM - A PERFORMANCE OPTIMIZING COMPUTER PROGRAM FOR DYNAMIC SYSTEMS SUBJECT TO TRANSIENT LOADINGS

by W. D. Pilkey, B. P. Wang, Y. Yoo, and B. Clark

Prepared by
UNIVERSITY OF VIRGINIA
Charlottesville, Va. 22901
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1973

1. Report No. NASA CR-2268	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle PERFORM - A PERFORMANCE OPTIMIZING COMPUTER PROGRAM FOR DYNAMIC SYSTEMS SUBJECT TO TRANSIENT LOADINGS		5. Report Date June 1973
		6. Performing Organization Code
7. Author(s) W. D. Pilkey, B. P. Wang, Y. Yoo, and B. Clark		8. Performing Organization Report No. AEEP-4085-105-71U
		10. Work Unit No.
9. Performing Organization Name and Address School of Engineering and Applied Science University of Virginia Charlottesville, VA 22901		11. Contract or Grant No. NGR 47-005-145
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

This report supplements material presented by the first two authors at the 42nd Shock and Vibration Symposium (Bulletin No. 42, 1971) and the 13th Structures, Structural Dynamics, and Materials Conference (AIAA Paper No. 72-340, April 10-12, 1972).

16. Abstract

This report contains a description and applications of a computer capability for determining the ultimate optimal behavior of a dynamically loaded structural-mechanical system. This capability provides characteristics of the "theoretically best," or limiting, design concept according to response criteria dictated by design requirements. Equations of motion of the system in first or second order form include incompletely specified elements whose characteristics are determined in the optimization of one or more performance indices subject to the response criteria in the form of constraints. The system is subject to deterministic transient inputs, and the computer capability is designed to operate with a large linear programming on-the-shelf software package which performs the desired optimization. The report contains user-oriented program documentation in engineering, problem-oriented form. Applications cover a wide variety of dynamics problems including those associated with such diverse configurations as a missile-silo system, impacting freight cars, and an aircraft ride control system.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Structural dynamics Optimization of dynamic systems	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	231	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

Page Intentionally Left Bl

FOREWORD

The research described in this report (University of Virginia Project No. 4085-0320) was performed at the University of Virginia Research Laboratories for the Engineering Sciences. W. D. Pilkey was principal investigator. The project was performed under NASA Grant No. NGR 47-005-145 and administered at NASA Langley Research Center.

Special thanks are due to Murray Manufacturing, Division Arrow Hart, Inc. of Earlysville, Virginia, who permitted the use of their computer terminal for communication with a computer center possessing software essential to this project. In addition Murray Manufacturing provided the special accessories necessary to adapt their terminal for our use.

Page Intentionally Left Bl

CONTENTS

SECTION I

INTRODUCTION

The current state of the design of physical systems is in some respects as much an art as a science. Of the many considerations involved in systems design, perhaps those related to an evaluation of the physical performance best lend themselves to formal automation. This report describes the results of an effort to place this aspect of real-world systems design on an analytical basis. In particular, a computational capability, PERFORM, has been developed for the evaluation of the limiting physical performance of systems subject to transient disturbances.

PERFORM provides the characteristics of the theoretically best, that is, the *limiting*, design concept according to response criteria. As a consequence, for certain mechanical, structural, and control systems PERFORM makes it possible to approach a design directly from the design criteria with no *a priori* commitment to a particular design concept. With this limiting performance capability, the designer can determine the feasibility of his proposed design on the basis of the specifications alone; moreover, he can monitor and measure his success during the design process itself. Heretofore, without the characteristics of the limiting design, the evaluation of proposed designs could be made only by performing a multitude of analyses for each candidate design.

This capability applies to systems with transient loading in which response criteria are of concern. These criteria are usually expressed in terms of *constraints* formed from displacements, stresses, forces, velocities, or accelerations that cannot exceed certain prescribed values. In addition, *objective functions* of peak values of other responses are to be minimized. PERFORM provides the time-optimal characteristics of portions of a system such that these constraints are satisfied and objective functions are *minimized* or *maximized*.

An outline of the capabilities of PERFORM is given in Section II.

Section III of this report describes the components and design of PERFORM. Also, the procedure for using PERFORM is outlined.

Section IV contains applications of PERFORM, including some numerical results. Among the problems discussed are rail vehicle suspension systems, train impact, aircraft ride control, launch vehicle control, missile/silo isolation, and a

nuclear reactor control system.

The Users Guide for PERFORM is given in Appendix I. The details of the technical formulation underlying PERFORM are in Appendix II. Appendix III contains the programming aspects of the system, including programming documentation. The final Appendix contains listings of programs.

SECTION II

CAPABILITIES OF PERFORM

PERFORM is a computer system that can be used to determine the limiting performance characteristics of a dynamic system subject to transient loading. Ref. 1 considers in detail the concept of limiting performance and its application to shock isolation systems. The example problems of Section IV provide an indication of the range of applications of PERFORM.

The dynamic system can be described and input to PERFORM using the first or second order equations

$$\dot{\bar{s}} = \underline{A}\bar{s} + \underline{B}\bar{u} + \underline{D}\bar{f}_k$$

$$\underline{M}\ddot{\bar{q}} + \underline{C}\dot{\bar{q}} + \underline{K}\bar{q} + \underline{U}\bar{u} = \underline{F}\bar{f}_k$$

in which $\bar{u}$ is a vector of time varying functions, called control or isolator forces, that have replaced portions of the dynamic system. $\underline{A}$, $\underline{B}$, $\underline{D}$, $\underline{M}$, $\underline{C}$, $\underline{K}$, $\underline{U}$, $\underline{F}$ are coefficient mattices. $\bar{s}$ and $\bar{q}$ are vectors of response varables, e.g., displacements, stresses, accelerations. $\bar{f}_k$ is a forcing function vector where the subscript k designates the k^{th} set of forcing or loading functions. This allows the system to encounter alternative sets of disturbances which might occur with equal probability.

The acceptable equations of motion appear to be linear. In fact, however, they are "quasilinear" since those portions of the system replaced by $\bar{u}$ can be linear, nonlinear, active, or passive. The remainder of the system must be linear as must the overall kinematics.

The user must place his equations in one of the forms of the above equations. The non-zero elements of the matrices $\underline{A}$, $\underline{B}$, $\underline{D}$ or $\underline{M}$, $\underline{C}$, $\underline{K}$, $\underline{U}$, $\underline{F}$ are then entered as inputs. This is accomplished by identifying the matrix, e.g., M MATRIX, and then specifying an element and its value, e.g., i, j, and M_{ij}. Elements not entered are assumed to be zero.

PERFORM finds the characteristics, including $\bar{u}$ and tradeoffs between optimal response variables, of the dynamic system such that bounds on some of the response variables $\bar{s}$ or $\bar{q}$ or control forces $\bar{u}$ are not violated while the maximum (or minimum) in time of other elements of $\bar{s}$ or $\bar{q}$ are minimized (or maximized).

Regardless of the form (first or second order) used to describe the equations of motion, the formats for the objective function and constraints are the same. In the case of the second order equations, a state variable vector $\bar{s}$ is established as

$$\bar{s} = \begin{bmatrix} \dot{\bar{q}} \\ \hline \bar{q} \end{bmatrix}$$

Any linear combination of state variables, derivatives of state variables, or control forces can be used as an objective function. In the case of the system described by second order equations, these become linear combinations of accelerations, velocities, displacements, and control forces. The objective function is input to PERFORM in the form of

$$\underline{PX1}\bar{s} + \underline{PX2}\bar{u} + \underline{PX3}\bar{f}_k$$

where $\underline{PX1}$, $\underline{PX2}$ and $\underline{PX3}$ are coefficient matrices. If more than one row of the matrices of this equation contains non-zero elements, then the peak values in time of the vectors resulting from the meaningful rows are to be compared. PERFORM minimizes (maximizes) the maximum (minimum) of the peak values.

Constraints may be placed on state variables, derivatives of state variables, and control forces. The general form, which is again linear, is

$$\overline{YL} \leq \underline{Y1}\bar{s} + \underline{Y2}\bar{u} + \underline{Y3}\bar{f}_k \leq \overline{YU}$$

where $\underline{Y1}$, $\underline{Y2}$, $\underline{Y3}$ are coefficient matrices and $\overline{YL}$, $\overline{YU}$ are lower and upper bound vectors. Constraints can be imposed at every time of the response or at specific times.

In summary, PERFORM accepts system equations of the form given above. For prescribed initial conditions, PERFORM then computes the $\bar{u}$ vector such that the max $|\underline{PX1}\bar{s} + \underline{PX2}\bar{u} + \underline{PX3}\bar{f}_k|$

is minimized (or min $|\ |$ is maximized) while the above constraints are satisfied. Any linear combination of $\bar{s}$, $\bar{u}$, and $\bar{f}_k$ can be tabulated or plotted versus time. A tradeoff curve between the maximum objective function and any particular constraint can be generated by varying the bounds on that constraint.

SECTION III

PERFORM SYSTEM DESCRIPTION

To do the limiting performance problem calculations a computer system named PERFORM has been developed. Because several separate programs are used, it is referred to as a system rather than a program.

In this section the system is described with increasing detail. First, the system is presented from the "systems analyst" point of view; the programs, data sets, and interrelations thereof are described and graphically presented with a system flowchart. Next the functions, subroutine calls, logic, etc. of each of the system programs are described. Program listings and discussions are in Appendix III.

A. System Design

PERFORM is a system of several programs, the functions, inputs, and outputs of which are outlined here. Influencing considerations of the development and operation environment are also discussed.

1. System Flow Description

Figure 1 is the PERFORM system flowchart. The three separate system programs are represented by rectangular boxes; these programs are:

- PREPROC (Pre-processor)
- LP Solver (linear programming computer program)
- PSTPROC (Post-processor)

a. PREPROC

Using the mathematical model of the dynamic system, the pre-processor program computes and punches the linear programming problem from the PERFORM problem specifications described in Appendix I. There are three outputs of this program:

- PSTPROC Report Specification. This card file contains information necessary for PSTPROC (to which this is an input file) to determine and print the problem solution.

- Linear Programming Problem Input. This card output is the linear pro-

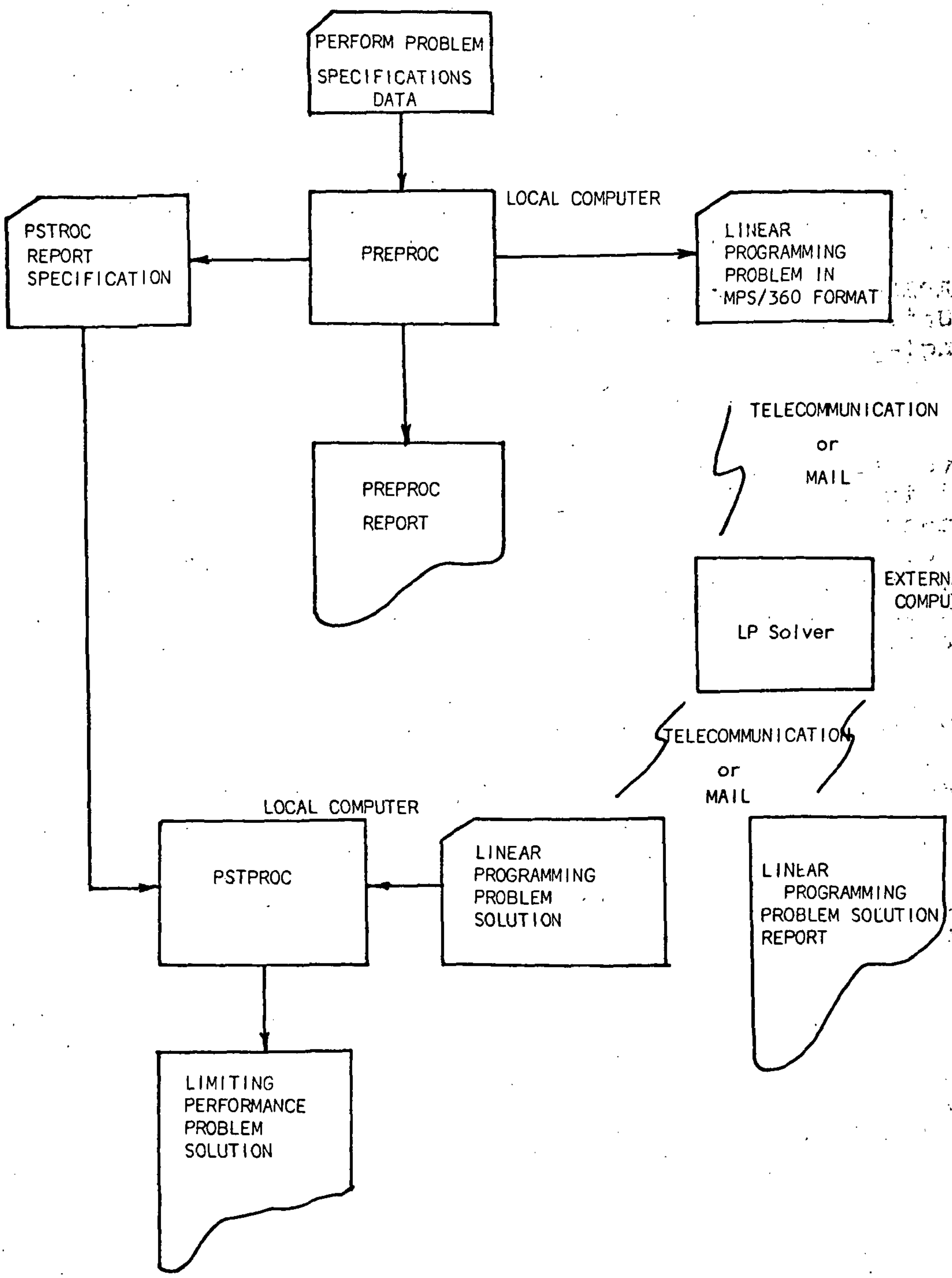


Fig. 1 PERFORM System Flowchart

gramming problem input in MPS/360 format, which is accepted by most major LP solvers.

PREPROC Report. This report is a printed record of the above files.

User options can control the contents of this report. The curved-bottom box is the standard flowcharting symbol for printed reports

The data transmission links are restricted somewhat in both the University of Virginia development environment and the NASA/Langley operating environment. These constraints are discussed in the section entitled "Environment Considerations."

PREPROC is written entirely in FORTRAN and has been tested on the University of Virginia Burroughs B5500,CDC6400, and the NASA/Langley CDC 6600.

b. LP Solver

This refers to any software package that can solve large linear programming problems. Examples of such capabilities are IBM MPS/360, CDC OPTIMA, and CDC OPHELIE. All of these programs accept MPS/360 input format. Since MPS/360 was used in the developmental stages of PERFORM, the LP Solver in the subsequent discussions of this report will be referred to as MPS/360. Anywhere MPS/360 is mentioned, other software packages, such as OPTIMA, could also be used.

MPS/360, Mathematical Programming System/360, is an open ended IBM software package capable of efficiently solving extremely large linear programming problems. The latest versions can handle problems of up to 16,000 rows and essentially unlimited columns. Several workfiles used by MPS/360 are omitted for clarity on the system flowchart of Fig. 1.

The input to MPS/360 is the output of "linear programming input" of PREPROC.

c. PSTPROC

The post-processor program produces the final report of the problem solution. Two inputs are necessary for PSTPROC: the linear programming problem solution from LP Solver and the PSTPROC report specifications from PREPROC. This program is written entirely in FORTRAN.

2. Environment Considerations

In Fig. 1 the programs are labeled "local" and "external" computer and a communications link is shown to and from the LP Solver step. This clearly awkward arrangement is used because

of the limitations of available computing facilities. The Langley Research Center CDC 6600 computing facility has no appropriate linear programming software. At Langley the PERFORM system cannot be run entirely locally, thus the need for an external computer. The system developed is a synthesis of these above constraints and the desire for maximum utilization of local facilities.

Punched cards are indicated as the storage medium for program data interchange. Cards were convenient for transmission to the external computer during development. Replacement with tape would be desirable and easily accomplished for large problems with unmanagable card files.

The decision to use MPS/360 was based on economy and availability. For development, the Virginia Polytechnic Institute and State University IBM/360 50-65 multiprocessor system, with an IBM 2780 telecommunications terminal was used. Recently, OPTIMA on the University of Virginia CDC 6400 has been used.

Use of the system is much simpler if a linear programming package is available locally. The data can be passed between programs on a disk, and the user does not have to be concerned with this data. One can simply put his problem specification data with some prepared control cards and receive the problem solution with one run.*

B. Program Descriptions

Each of the three programs of the system is further described in this section. The purpose of the description is to relate the program coding to the limiting performance problem model but without involving computational details. A detailed listing with more discussion is given in Appendix III.

1. PREPROC

Figure 2 is a flowchart of the main section of PREPROC. This section contains all input, calls to computational subroutines, and varying bound calculations.

Some notes about the flowchart are in order. Predefined blocks, the blocks with outward pointed sides and parallel top and bottom, represent subroutine calls; the name in the block is that of the subroutine. The trapezoidal-shaped box enclosing most of the third page and beneath the label "For each varying bound" is unconventional. It signifies that the sequence of instructions in the block is to be executed for the

* This is currently possible on an automated version of PERFORM at the University of Virginia Computer Science Center, both for on-campus users and for users accessing the computer from a remote, compatible computer terminal.

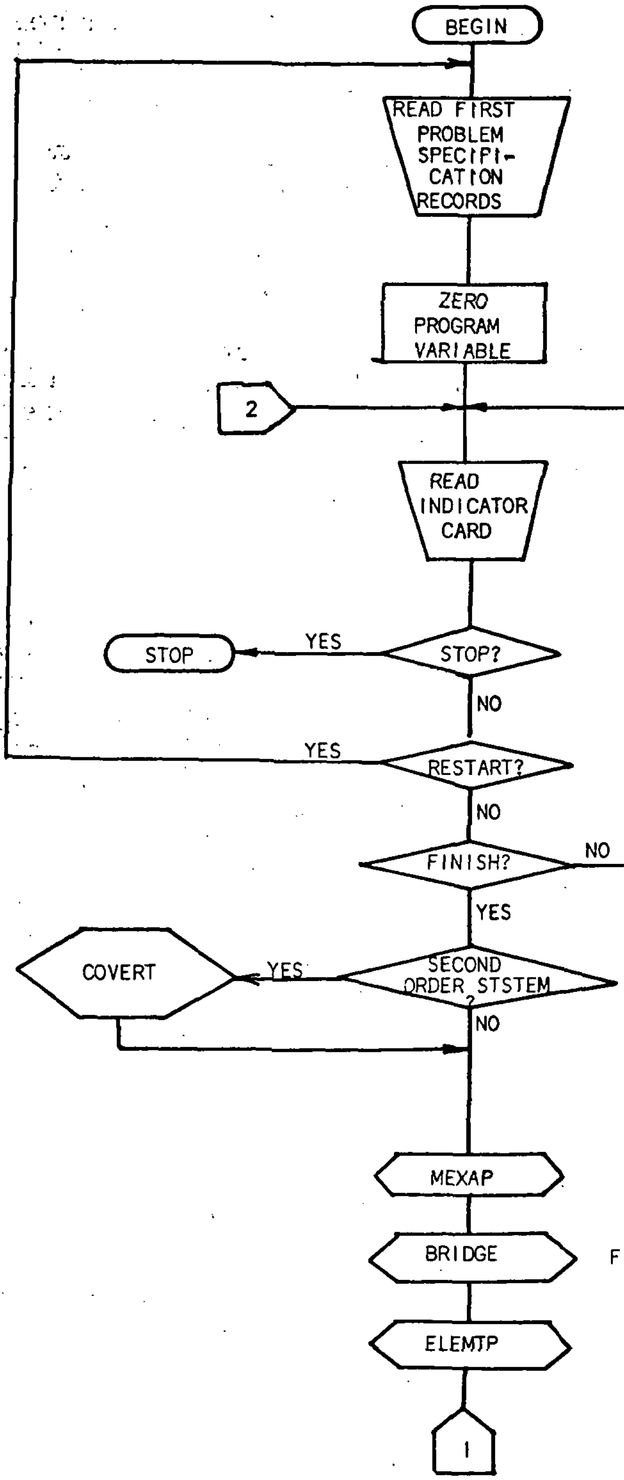


Fig. 2. PREPROC Main Section

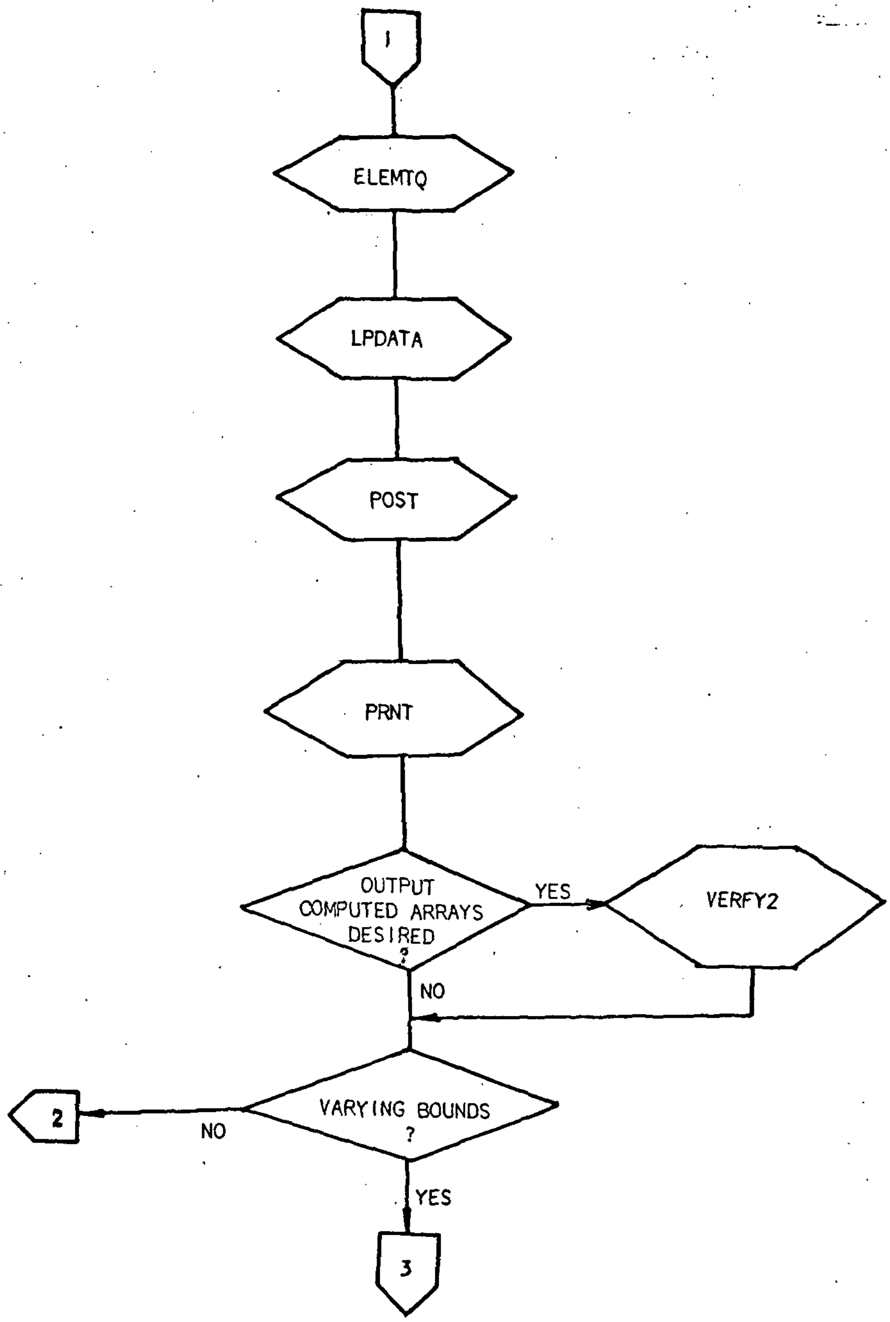


Fig. 2 PREPROC Main Section (continued)

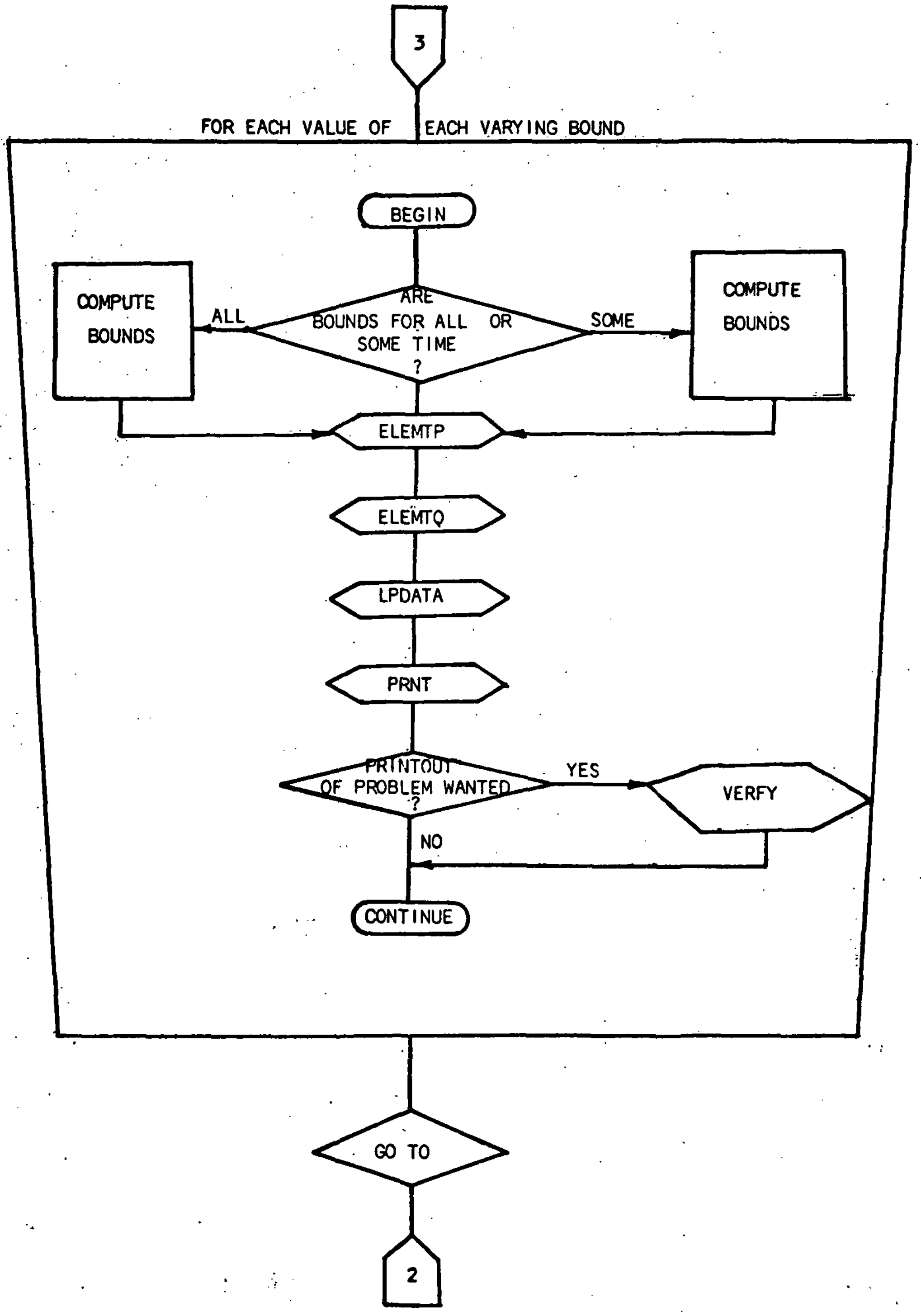


Fig. 2 PREPROC Main Section (concluded)

value of each varying bound, i.e., varying row of the $\overline{YL}$ and $\overline{YU}$ vectors. This flowchart is not an exact representation of the actual program; from it one can tell what the program does but not necessarily how it is done.

Because all files except those for printed output in the PERFORM system are either cards or in card format, the word "record" used to describe file elements can be interpreted as "card" in this discussion. The records of the problem specifications file are of two types: data and indicator. Data records contain values of program variables. By preceding data records, indicator records associate values with variables; indicator cards also contain problem processing specifications.

Before each section of data records of values for program variables there is an indicator card specifying the program variable array. The last card of a section contains a 1 in the key field; previous cards of the section have a zero in this field. Following the last card of a section the program presumes an indicator card. The program contains no error detection facilities so it behooves the user to double check his input. Details of these records are found in Appendix I. The first thing done by the program is to read three problem specifications records; in Appendix I these three records are numbered cards 1, 2, and 3. Next, most program variables are initialized to zero.

The fourth block reads an indicator card. If the card is a STOP indicator card the program stops. If a RESTART indicator, control returns to the program beginning and a new problem may be processed. A FINISH INPUT DATA record starts the problem processing. Appropriate processing for other indicators is performed - this includes such things as array entry, array printing, problem modification, etc.

If the dynamic system equations are second order, subroutine COVERT is called to reduce the second order equations to first order. The order reduction method is described in Appendix II.

Subroutine MEXAP computes exponential functions of matrices. These are array $\underline{P}$ and array $\underline{AEI}$, both defined in Appendix II.

The next routine, BRIDGE, computes arrays for matrices $\underline{R}$, $\underline{T}$, and vectors $\overline{C1}$, $\overline{C2}$, and $\overline{C}$ as described in Appendix II.

Routine ELEMTP is called to compute those portions of the $\underline{H}$ and $\overline{G}$ arrays that are determined by the objective function. These portions are shown in Table 1 of Appendix II.

The other portions of these arrays, determined by the problem constraints, as shown in Tables 2A and 2B of Appendix II, are calculated in subroutine ELEMTQ. Subroutines ELEMTR, ELEMTS, and ELEMTT are called by ELEMTQ for calculations but are not shown in Fig. 2. With $\underline{H}$ and $\overline{G}$ arrays computed, the next step is to output the results.

Subroutine LPDATA is called to produce the MPS/360 linear programming problem. Other subroutines, not shown, are called to punch different sections of the MPS input: CODATA for columns, RODATA for rows, RHDATA for the right hand sides, and BODATA on the bounds.

The POST routine punches the PSTPROC report specifications. Although this file is shown separately from the MPS/360 linear programming file the user must separate the two from the single card punch output.

To provide a printed problem record subroutine PRNT prints the $\underline{H}$ and $\overline{G}$ arrays. If a printout of the calculated arrays is desired, as indicated by a nonzero value of variable INT in the VERIFY section, it is produced by subroutine VERFY. The problem specification is produced when the VERFY section is processed and variable INP is nonzero.

Any number of the lower or upper bounds of the rows of the matrix expression $\overline{YU} \leq \cdot \cdot \cdot \cdot \cdot \cdot \leq \overline{YL}$ may be varied. Appendix I contains details of how this is specified.

For each value of each varying bound the linear programming problem is computed, punched, and printed by the subroutine sequence ELEMTP, ELEMTQ, LPDATA, and PRNT. If a printout of the problem arrays is desired as indicated by a nonzero value of variable INT, it is produced by subroutine VERFY. When the loop has been performed for each value of each varying bound, control returns to the indicator point for modifications or other instructions.

There are several additional array manipulation subroutines in PREPROC not mentioned here. These are described in Appendix III A.

2. MPS/360

Mathematical Programming System/360 is an open ended IBM software package for solving optimization problems. MPS is not a program, but a set of procedures with linking facilities. The procedures can do such things as read data, set up files, write solutions, and, of course, solve linear programming problems. The procedure linking facility is known as the control program. It consists of procedure call statements

in addition to ordinary programming facilities such as GOTO, IF, MOVE statements, etc. This control program is compiled and executed in separate OS job steps.

There are two outputs from MPS: the printed linear programming problem report and the punched problem solution.

If the punched linear programming problem output of PREPROC is interpreted, a similarity to the PREPROC input can be observed. Nonzero program variables are entered on cards with array row and column numbers and follow section identifying cards. Unlike PREPROC the alpha characters ROW, COL precede the left justified numbers. The characters are necessary because MPS identifies rows and columns of arrays with FORTRAN-like variables which must begin with an alphabetical character. ROW and COL serve to identify columns of the linear programming tableau; RHS identifies the right hand side vector and BDS identifies the variable bounds specifications.

The control program of MPS/360 for PERFORM is rather simple; the data is read in, the problem solved, and the solution written and punched. A listing and discussion of the control program is in Appendix IIIB.

Each linear programming problem produced for each value of each varying bound requires a separate MPS/360 run.

3. PSTPROC

Program PSTPROC writes the solution to the limiting performance problem using the MPS/360 linear programming problem solution and the PSTPROC report data from PREPROC.

Figure 3 is a functional flowchart of PSTPROC. First the PSTPROC report data, punched by subroutine POST in program PREPROC, is read. Next the first linear programming problem solution is read. Again it should be pointed out that no error detection facilities are in the program. If a card is dropped from the PSTPROC report specifications a linear programming solution card will take its place.

For each row of the tabulation expression, i.e., the Q1, Q2, and Q3 expression, the block inside the box is executed. For all time intervals the value of the expression is calculated and printed. If a graphical presentation of these calculations is desired, as specified by a nonzero value ITR(I) where I is the row number in Q1, Q2, Q3 matrix expression, subroutine TRAJ is called.

After all the tabulation expressions have been procured

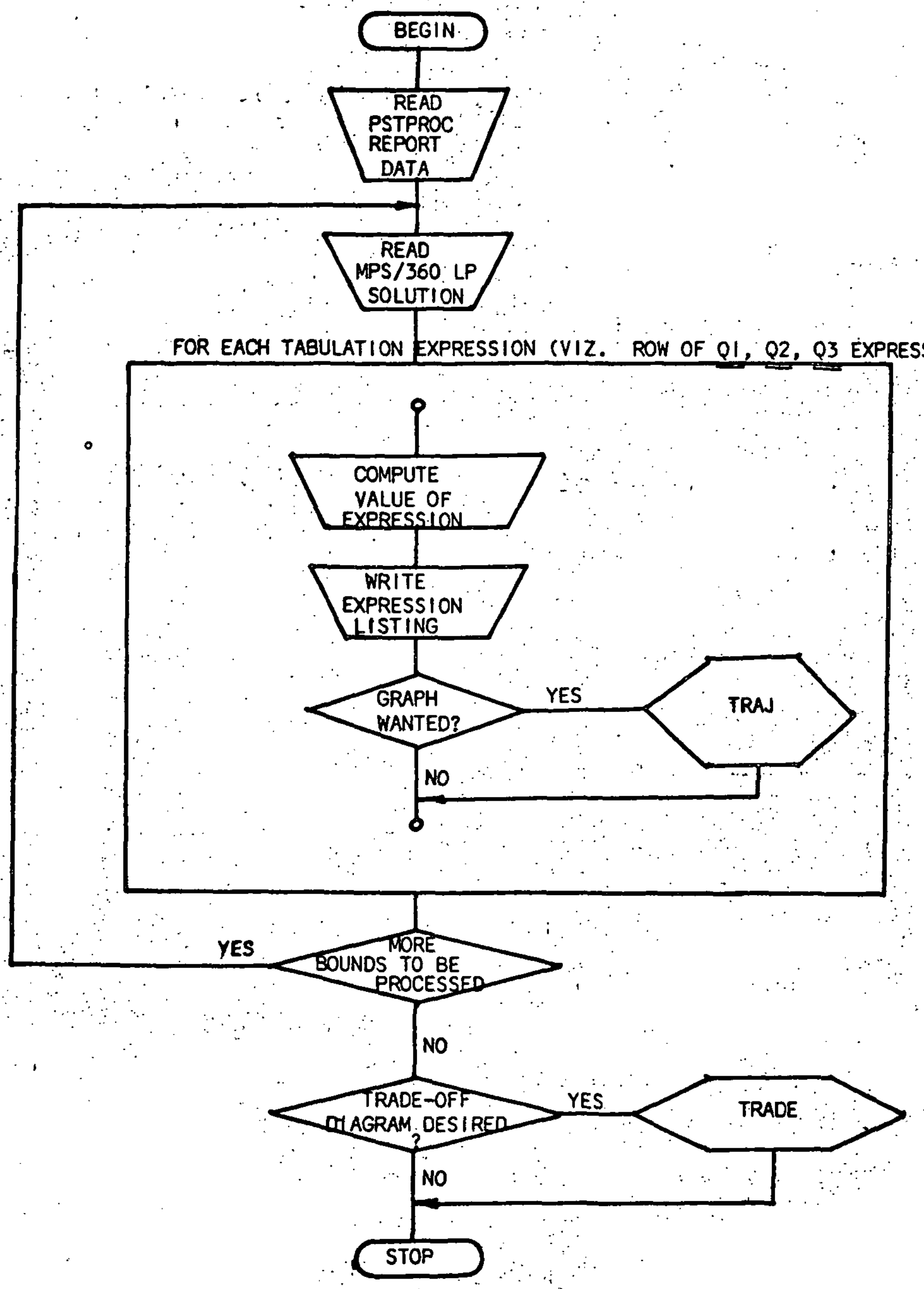


Fig. 3 PSTPROC Main Section, Functional Fow Chart

the next linear programming solution is read and the calculation loop is repeated. If all solutions have been processed variable ITD is checked.

If ITD is nonzero, a trade-off diagram is printed by subroutine TRADE.

After the trade-off diagram is produced processing stops. Note that PSTPROC handles only one problem a run; different problems, with different PSTPROC report decks, require separate runs.

SECTION IV

APPLICATIONS

A. Single-Degree-of-Freedom Shock Isolation System

We begin the applications of PERFORM by considering a single-degree-of-freedom (SDF) dynamic system subject to transient loading. Since this system has been thoroughly explored elsewhere (e.g., Ref. 1), it is included here only to demonstrate the use of PERFORM in the study of the limiting performance of a familiar system.

Consider the SDF system together with the acceleration shock shown in Fig. 4. The equation of motion is given by

$$m\ddot{z} + u = 0 \tag{1}$$

or, for a unit mass,

$$\ddot{z} + u = 0 \tag{2}$$

with initial conditions $\dot{z}(0) = z(0) = 0$

The kinematic relations are

$$z = x + y \tag{3a}$$

$$\dot{z} = \dot{x} + \dot{y} \tag{3b}$$

$$\ddot{z} = \ddot{x} + \ddot{y} = \ddot{x} + f \tag{3c}$$

Substituting Eq(3c) into Eq(2) gives

$$\ddot{x} + u = -\ddot{y} = -f$$

$$x(0) = y(0) = 0$$

$$\dot{x}(0) = \dot{y}(0) = 0$$

Suppose we wish to find the lowest possible peak acceleration of the mass ($\ddot{z}$) if the relative displacement between the mass and the base is constrained. The function u(t) is the system controller or the isolator function. Thus, we wish to find u(t) such that the maximum acceleration of the mass

$$\phi = \max |\ddot{z}| = \max |u|$$

is minimized while the relative displacement satisfies the constraint

$$|x| \le A$$

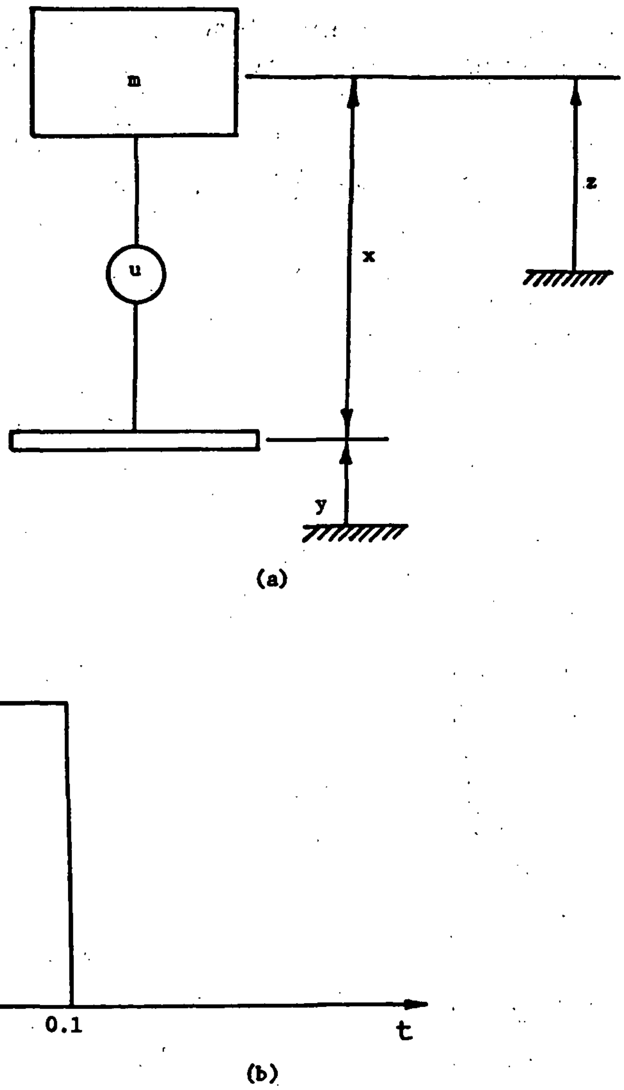


Fig. 4 A SDF System with Acceleration Shock Input

We seek the trade-off between minimum ϕ and a range of values of A. The design value of this information is discussed in Ref. 1.

Specifically, we choose to compute min ϕ for A = 0.01, 0.02, and 0.03.

The formulation of the problem in the PERFORM format follows. The notation is defined in Appendix I.

1. Equations of Motion ($\ddot{x} + u = -f$)

$$\underline{M}\ddot{\bar{q}} + \underline{C}\dot{\bar{q}} + \underline{K}\bar{q} + \underline{U}\bar{u} = \underline{F}\bar{f}_k$$

where

$$\underline{M} = [1]$$

$$\underline{C} = [0]$$

$$\underline{K} = [0]$$

$$\underline{U} = [1]$$

$$\underline{F} = [-1]$$

$$\bar{q} = [x]$$

$$\dot{\bar{q}} = [\dot{x}]$$

$$\ddot{\bar{q}} = [\ddot{x}]$$

$$\bar{u} = [u]$$

$$\bar{f}_k = [\ddot{y}] = [f]$$

2. Forcing Function (Fig. 4b)

$$f(1,j,1) = 10.0 \text{ for } j = 1 \text{ to } 5$$

$$f(1,j,1) = 0 \text{ for } j = 6 \text{ to } 9$$

where the first index 1 designates that this is the first (and the only) set of forcing functions, j = discrete time, the last index 1 indicates the first component in forcing function set No. 1.

3. Objective Function (ϕ)

$$\underline{PX1}\ \bar{s} + \underline{PX2}\ \bar{u} + \underline{PX3}\ \bar{f}_k$$

where

$$\underline{PX1} = [0\ 0]$$

$$\bar{s} = \begin{bmatrix} \dot{x} \\ x \end{bmatrix}$$

$$\underline{PX2} = [1]$$

$$\underline{PX3} = [0]$$

4. <u>Constraints</u> ($|x| \leq A$)

$$\overline{YL} \leq \underline{Y1}\bar{s} + \underline{Y2}\bar{u} + \underline{Y3}\bar{f}_k \leq \overline{YU}$$

where

$$\underline{Y1} = [0\ 1]$$

$$\underline{Y2} = [0]$$

$$\underline{Y3} = [0]$$

Since the constraint is for all time and of ($\leq\ \leq$) type we set

MSP(1,1) = 0

MSP(1,2) = 0

To obtain a trade-off diagram the value of A is varied. Thus

$$\overline{YU} = A_{01} + n\Delta A_1$$

$$\overline{YL} = A_{02} + n\Delta A_2$$

where

$$A_{01} = SVA1 = 0.1$$

$$A_{02} = SVA2 = -0.1$$

$$\Delta A_1 = 0.1$$

$$\Delta A_2 = -0.1$$

$$n = 1 \text{ to } NIC$$

5. Output

The time trajectory of u is obtained using the output matrix

$$\underline{Q1}\bar{s} + \underline{Q2}\bar{u} + \underline{Q3}\bar{f}_k$$

$$\underline{Q1} = [0 \;\; 0]$$

$$\underline{Q2} = [1]$$

$$\underline{Q3} = [0]$$

6. Other Data Pertinent to This Problem

We set

NOE = 2

MODE = 1

NDF = 1

NU = 1

NF = 1

NSETS = 1

II = 10

TM = 0.02

NOB = 1

NOC = 1

ISP = 0

NOT = 1 (here we assume only one output is required)

and

ISET(1) = 1

The whole input deck is shown in Fig. 5.

```
          1    SINGLE-DEGREE-OF-FREEDOM SYSTEM
          2    0         1    1    1    1   10  0.02        1   -1    0    3
                    1
     M MATRIX
     1    1    1    1.0
     F MATRIX
     1    1    1   -1.0
     U MATRIX
     1    1    1    1.0
     FORCING FUNCTION
     1    1
     1    1    0
     0    1        10.0
     0    2        10.0
     0    3        10.0
     0    4        10.0
     1    5        10.0
     PX2 MATRIX
     1    1    1    1.0
     Y1 MATRIX
     1    1    2    1.0
     BOUNDS OF CONSTRAINT
     1    1    0    0
      0
        0.01           2     0.01
      0
       -0.01           2    -0.01
     Q2 MATRIX
     1    1    1     1.0
     TRAJECTORY
     0    1    1
     0    2    1
     1    3    1
     TRADE OFF DIAGMAM
         1    1
     VERIFY
         1    1
     FINISH INPUT DATA
     STOP
```

Fig. 5. Input Deck of SDF System

7. Results

Figures 6a through 6g show the output of PERFORM resulting from the input deck of Fig. 5. These are the time trajectories of the control forces (u) (Figs. 6a to 6f) and a diagram (Fig. 6g) indicating the performance trade-off between minimum ϕ the objective function, and A, the bound of the constraint.

```
TB= Q1*S+ Q2*U+ Q3*F

  WHERE

    Q1=  0.00  0.00
    Q2=  1.00
    Q3=  0.00
```

TIME INTERVAL			TB
0.000	TO	.020	-8.333333
.020	TO	.040	-8.333333
.040	TO	.060	-8.333333
.060	TO	.080	-8.333333
.080	TO	.100	-8.333333
.100	TO	.120	-8.333333
.120	TO	.140	-8.333333
.140	TO	.160	0.000000
.160	TO	.180	0.000000

Fig. 6a. Trajectory of u(t) for $|x| \leq 0,01$.

```
TB= Q1*S+ Q2*U+ Q3*F

  WHERE

    Q1=  0.00  0.00
    Q2=  1.00
    Q3=  0.00
```

TIME INTERVAL			TB
0.000	TO	.020	-7.142857
.020	TO	.040	-7.142857
.040	TO	.060	-7.142857
.060	TO	.080	-7.142857
.080	TO	.100	-7.142857
.100	TO	.120	-7.142857
.120	TO	.140	-7.142857
.140	TO	.160	0.000000
.160	TO	.180	0.000000

Fig. 6b. Trajectory of u(t) for $|x| \leq 0.02$.

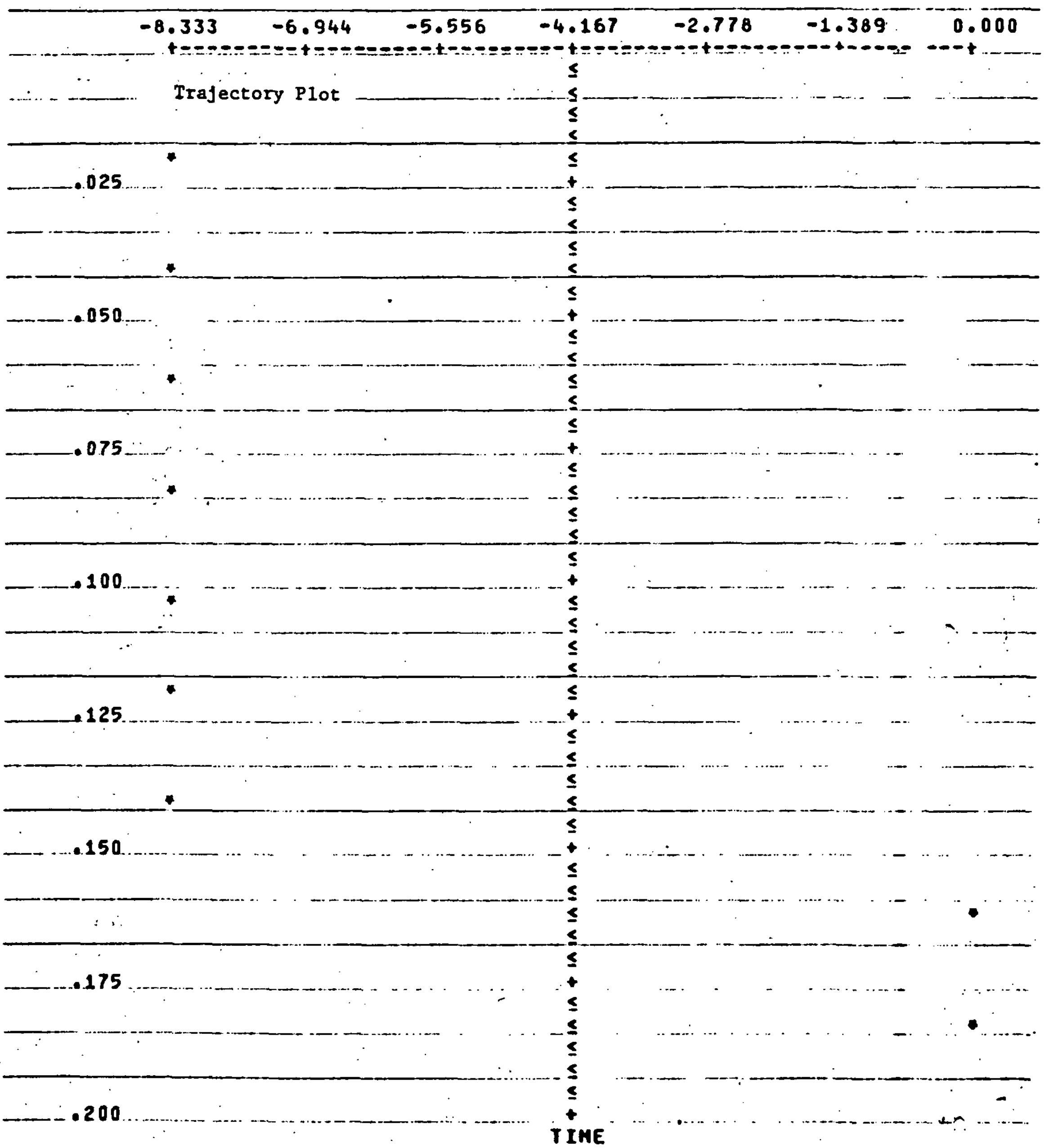


Fig. 6c. Plot of u(t) for $|x| \leq 0.01$.

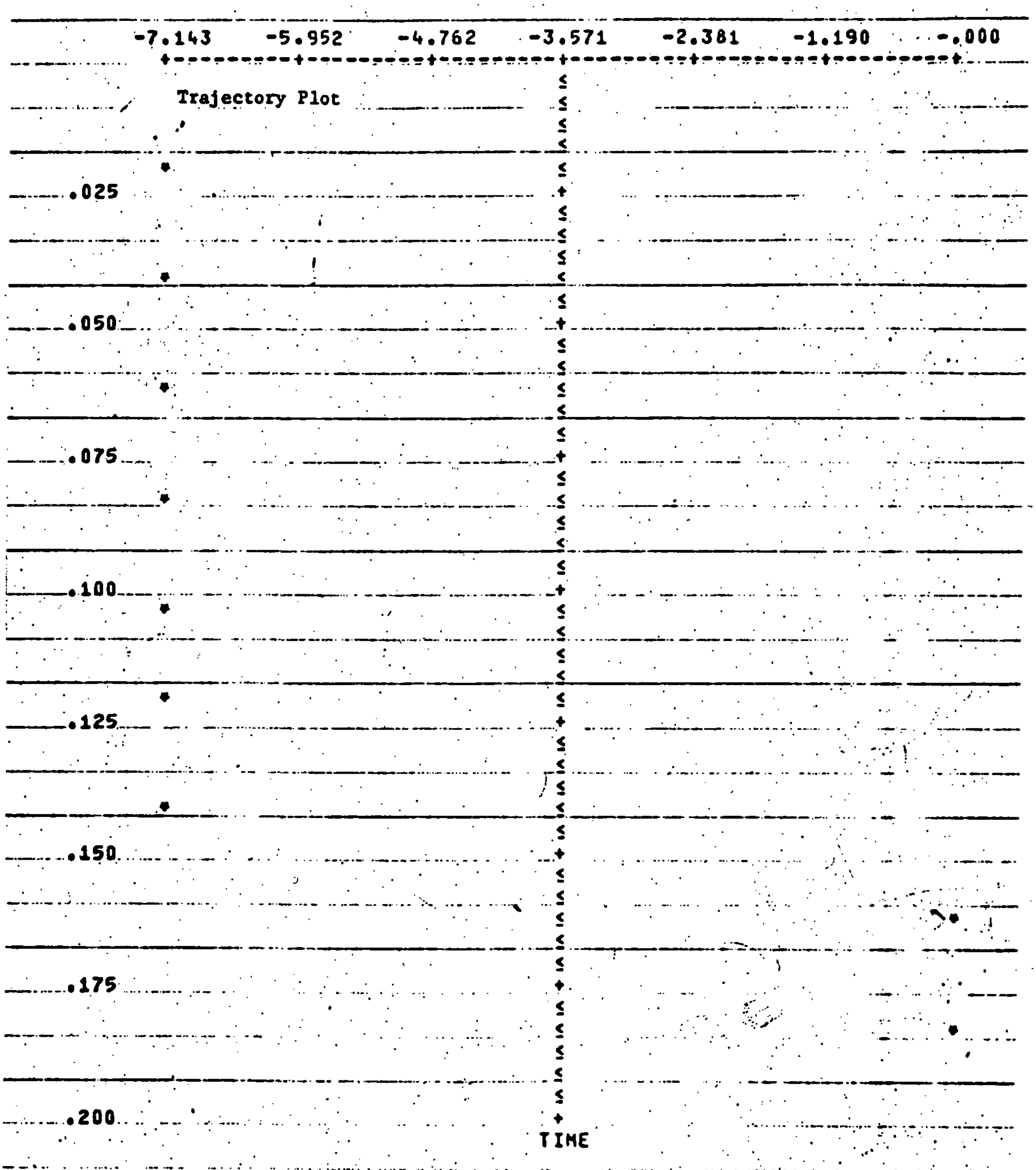


Fig. 6d. Plot of u(t) for $|x| \leq 0.02$.

TB= Q1*S+ Q2*U+ Q3*F

WHERE

Q1= 0.00 0.00
Q2= 1.00
Q3= 0.00

TIME INTERVAL			TB
0.000	TO	.020	-6.250000
.020	TO	.040	-6.250000
.040	TO	.060	-6.250000
.060	TO	.080	-6.250000
.080	TO	.100	-6.250000
.100	TO	.120	-6.250000
.120	TO	.140	-6.250000
.140	TO	.160	-6.250000
.160	TO	.180	0.000000

Fig. 6e. Trajectory of u(t) for $|x| \leq 0.03$.

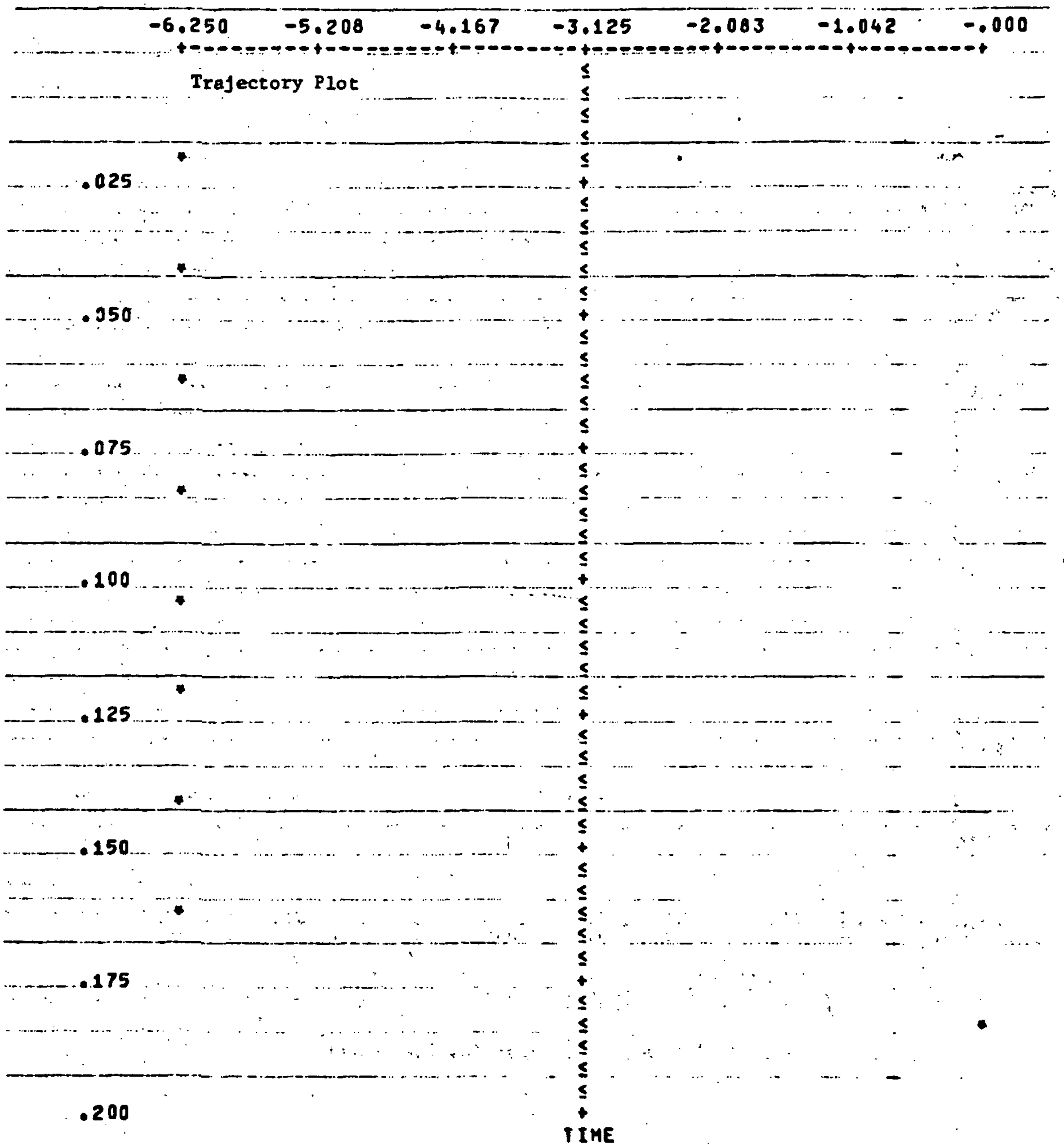
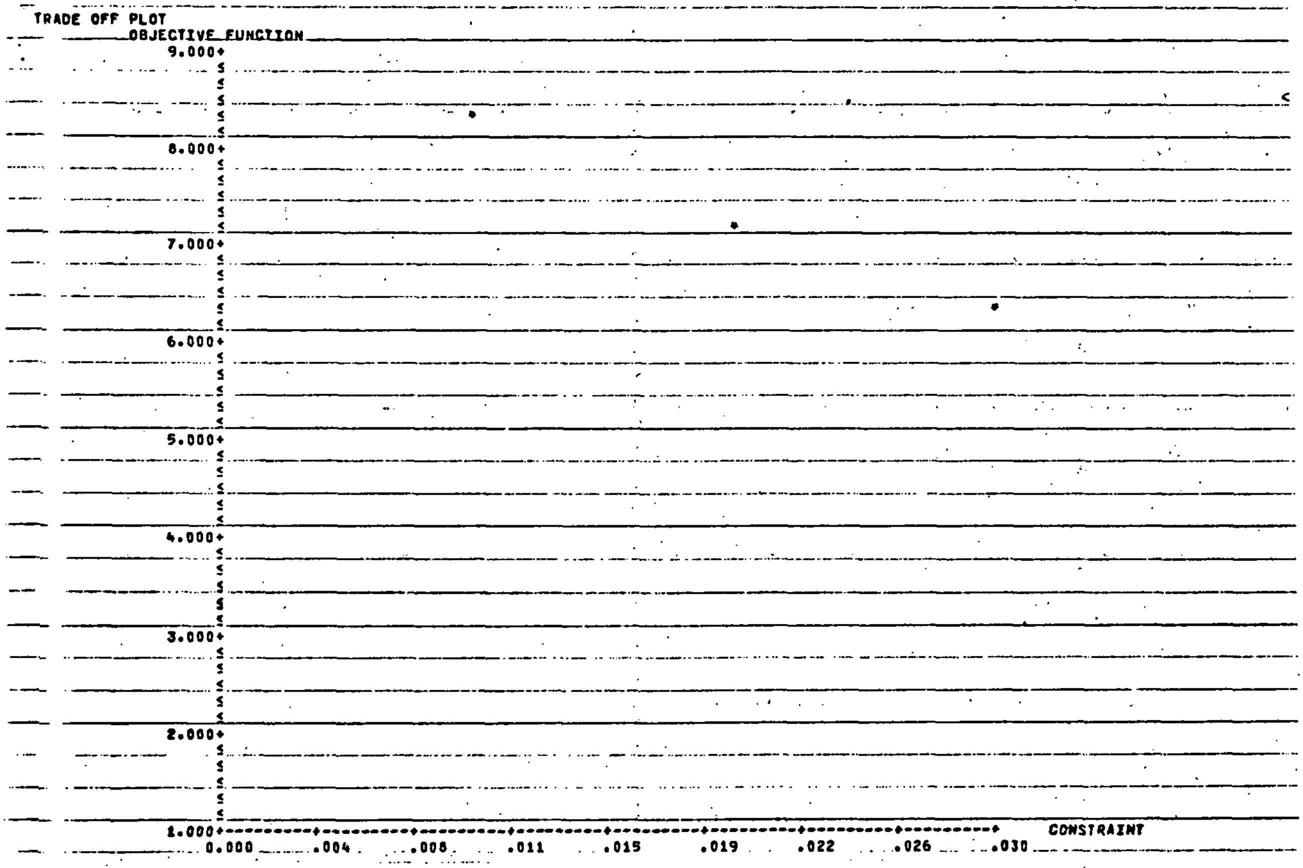


Fig. 6f. Plot of u(t) for $|x| \leq 0.03$.



OBJECTIVE FUNCTION = 8.333333 BOUND = .010000
OBJECTIVE FUNCTION = 7.142857 BOUND = .020000
OBJECTIVE FUNCTION = 6.250000 BOUND = .030000

Fig. 6g. Trade-off Diagram for SDF System

B. Multi-Degree-of-Freedom Shock Isolation Systems

To use PERFORM in evaluating the limiting performance characteristics of a shock isolation system the equations of motion of the system, after the control forces are introduced, must be linear. However, many dynamic systems are described by nonlinear equations and we are invariably faced with the problem of approximating the systems with various assumptions and yet not losing sight of the actual system. In this section, in addition to treating some multi-degree-of-freedom shock isolation systems, we will consider some of the reasonable assumptions to be made for the linearization of the equations of motion for some dynamic systems of nonlinear nature.

1. Klein's Three-Degree-of-Freedom System

a. Problem Description

In Ref. 2 a three-degree-of-freedom system as depicted in Fig. 7 was considered. The model to be analyzed has the z_1, z_2 plane as a plane of symmetry. It is assumed that the mass is a rigid body; that principal axes of inertia through the mass center parallel the edges of the body; and that the motion of the mass will not affect the motion of the base. In addition, it is assumed that the points of application of the shock isolator forces u_1, u_2, u_3 and u_4 remain fixed at the corners of the supported mass and retain their horizontal and vertical directions despite the motions of the base and mass. The kinematic conditions of the system are shown in Fig. 7, to which the following definitions apply:

u_i = the shock isolators or control forces which are to be optimized,

y_i = the two orthogonal input motions of the base referred to a fixed reference frame,

z_i = the two orthogonal motions of the center of mass relative to a fixed reference frame,

θ = rigid body rotation of the isolated mass about the center of mass.

The equations of motion for the system shown in Fig. 7 become:

$$\begin{aligned} m\ddot{z}_1 - u_1 + u_4 &= 0 \\ m\ddot{z}_2 - u_2 - u_3 &= 0 \end{aligned} \tag{4}$$

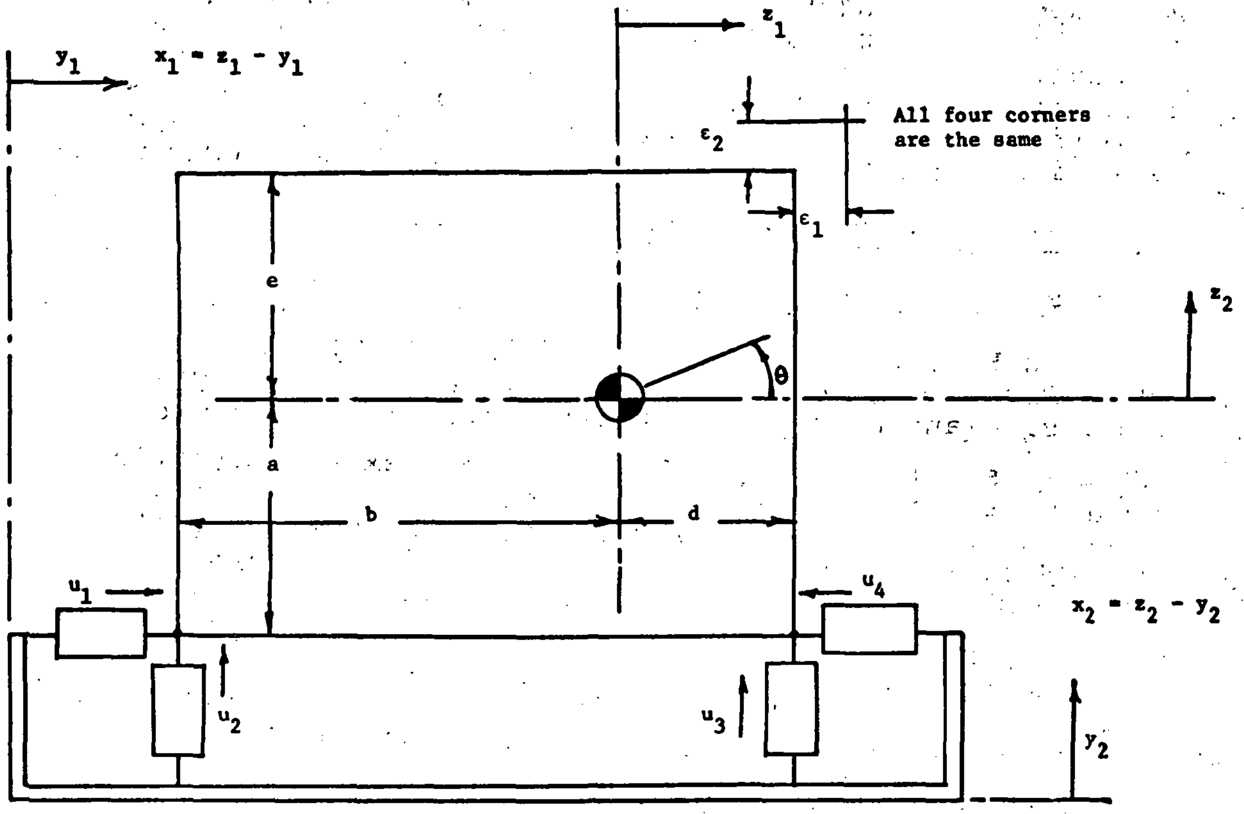


Fig. 7 Active Isolation System Used in Ref. 2

$$I\ddot{\theta} - au_1 \cos\theta + bu_2 \cos\theta - du_3 \cos\theta + au_4 \cos\theta = 0 \quad (4c)$$

where m is the isolated mass and I is the moment of inertia of the isolated mass about its center of mass.

It is further assumed in Ref. 2 that θ is small, therefore allowing the usual small angle approximations to be made. Using the kinematic relations

$$z_1 = x_1 + y_1, \; \ddot{z}_1 = \ddot{x}_1 + \ddot{y}_1$$

$$z_2 = x_2 + y_2, \; \ddot{z}_2 = \ddot{x}_2 + \ddot{y}_2$$

Eq. (4) becomes

$$\begin{aligned} m\ddot{x}_1 - u_1 + u_4 &= -m\ddot{y}_1 \\ m\ddot{x}_2 - u_2 - u_3 &= -m\ddot{y}_2 \\ I\ddot{\theta} - au_1 + bu_2 - du_3 + au_4 &= 0 \end{aligned} \quad (5)$$

Letting $I = m\rho^2$, where ρ is the radius of gyration, we find for unit mass

$$\begin{aligned} \ddot{x}_1 - u_1 + u_4 &= -\ddot{y}_1 \\ \ddot{x}_2 - u_2 - u_3 &= -\ddot{y}_2 \\ \ddot{\theta} - \frac{au_1}{\rho^2} + \frac{bu_2}{\rho^2} - \frac{du_3}{\rho^2} + \frac{au_4}{\rho^2} &= 0 \end{aligned} \quad (6)$$

The above equations define the behavior of the isolated mass as a function of the applied accelerations $\ddot{y}_1$ and $\ddot{y}_2$ and the control forces u_1, u_2, u_3, and u_4.

For this linearized system the optimization problem considered in Ref. 2 is: Find the control functions $u_1(t)$, $u_2(t)$, $u_3(t)$ and $u_4(t)$ so as to minimize some function of the maximum absolute values of the response acceleration, $f(|\ddot{z}_1|_{max}, |\ddot{z}_2|_{max}, |\ddot{\theta}|_{max})$ when the base is subjected to input $\ddot{y}_1$ and/or $\ddot{y}_2$ and the mass is constrained to move within the rattlespaces defined by ε_1 and ε_2 as shown in Fig. 7.

Since linear programming is selected as a solution technique, the objective function and constraints must be expressible as linear combinations of control forces. The objective function chosen was

$$\Phi = \max \{\max |w_1(t)|, \max |w_2(t)|, \max |w_3(t)|\} \tag{7}$$

where
$$\begin{aligned} w_1 &= \alpha_1(u_1 - u_4) = \alpha_1\ddot{z}_1 \\ w_2 &= \alpha_2(u_2 + u_3) = \alpha_1\ddot{z}_2 \\ w_3 &= \frac{a}{\rho^2} u_1 - \frac{b}{\rho^2} u_2 + \frac{d}{\rho^2} u_3 - \frac{a}{\rho^2} u_4 = \ddot{\theta} \end{aligned} \tag{8}$$

with α_1, α_2 as weighting constants.

The constraints are

$$\begin{aligned} |x_1 + a\theta| &\leq \varepsilon_1 \\ |x_1 - e\theta| &\leq \varepsilon_1 \\ |x_2 - b\theta| &\leq \varepsilon_2 \\ |x_2 + d\theta| &\leq \varepsilon_2 \end{aligned} \tag{9}$$

The forcing function considered was

$$y(t) = t^2e^{-t} \tag{10}$$

This gives an acceleration

$$\ddot{y}(t) = e^{-t}(t^2 - 4t + 2) \tag{11}$$

To compare the use of PERFORM with results available in Ref. 2, the forcing function is chosen to act in the y_1 direction only. Other pertinent data include:

No. of time intervals = 40

$$\alpha_1 = 1.0$$

$$\alpha_2 = 1.0$$

$$\varepsilon_1 = 0.1$$

$$\varepsilon_2 = 0.0$$

$$a = 0.0$$

$$b = 25.0$$

$$d = 25.0$$

$$e = 0.0$$

b. Formulation in PERFORM Format

The Ref. 2 example outlined above will now be reformulated in accordance with the PERFORM format. Since the forcing function acts in the y_1 direction, u_2 and u_3 will be inactive, as will w_2. Hence only two functions will be included in the objective function. Due to $\varepsilon_2 = 0$, only the constraints on x_1 will be taken into account.

Equations of Motion

Using the definitions for w_1, w_2, and w_3 as given in Eq. (8) and the values of the constraints as defined above, the equations of motion can be written as

$$\begin{aligned} \ddot{x}_1 - w_1 &= -\ddot{y}_1 \\ \ddot{x}_2 - w_2 &= 0 \\ \ddot{\theta} - w_3 &= 0 \end{aligned} \tag{12}$$

Eq. (12) may be rewritten in the PERFORM format as

$$\underline{M}\ddot{\bar{q}} + \underline{C}\dot{\bar{q}} + \underline{K}\bar{q} + \underline{U}\bar{u} = \underline{F}\bar{f}_k \tag{13}$$

where the vectors are defined as

$$\bar{q} = \begin{bmatrix} x_1 \\ x_2 \\ \theta \end{bmatrix} \qquad \dot{\bar{q}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\theta} \end{bmatrix} \qquad \ddot{\bar{q}} = \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{\theta} \end{bmatrix}$$

$$\bar{f}_k = [\ddot{y}_1]$$

$$\bar{u} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$$

and the coefficient matrices as

$$\underline{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{C} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{K} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\underline{U} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\underline{F} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

Objective Function

The objective function given by Eq. (7) is to be minimized. In the PERFORM input format the objective function becomes

$$\underline{PX1}\bar{s} + \underline{PX2}\bar{u} + \underline{PX3}\bar{f}_k \qquad (14)$$

where

$$\bar{s} = \begin{bmatrix} \dot{\bar{q}} \\ \bar{q} \end{bmatrix} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{\Theta} \\ x_1 \\ x_2 \\ \Theta \end{bmatrix}$$

and

$$\underline{PX1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{PX2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\underline{PX3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Constraints

The constraint is given on the relative displacements as

$$|x_1| \leq \varepsilon_1 \quad \text{or}$$

$$-\varepsilon_1 \leq x_1 \leq \varepsilon_1$$

In the PERFORM format this becomes

$$\overline{YL} \leq \underline{Y1}\ \bar{s} + \underline{Y2}\ \bar{u} + \underline{Y3}\ \bar{f}_k \leq \overline{YU} \qquad (15)$$

where

$$\overline{YL} = [-\varepsilon_1] = [-0.1]$$

$$\overline{YU} = [\ \varepsilon_1] = [\ 0.1]$$

$\underline{Y1} = [0\ 0\ 0\ 1\ 0\ 0]$

$\underline{Y2} = [0\ 0\ 0]$

$\underline{Y3} = [0]$

The input deck is shown in Fig. 8.

c. Comparison of Results

The results obtained by using a linear programming optimization procedure for this case given in Ref. 2 are duplicated in Table 1. The results of the same problem solved by PERFORM are given in Table 2a. The values of the objective function from these two solutions are very close to each other. The tabulated output from PERFORM is the w_1 in Ref. 2. Using the values of u_1 and u_4 from Table 1, the w_1 calculated from Eqs. (8) are about the same as those given by PERFORM for the first five time intervals. A comparison of the w_1 values for the first 10 time intervals is given in Table 2b. Since the solution of the optimization problem is not unique after the maximum value of the objective function has been reached, it is not surprising that Table 1 and Table 2a do not agree consistently at later times.

d. Modified System

Based on certain assumptions, the above system can be modified to include large rotations. Additional isolator forces u_α and u_β can be introduced for rotational motion of the isolated mass, and rotational contributions of horizontal and vertical isolator forces can be neglected (i.e., omission of all but $\ddot{\theta}$ in the third of Eqs. (6)). The equations of motion for the system shown in Fig. 9 are written as

$$\begin{aligned} \ddot{x}_1 - u_1 + u_4 &= -\ddot{y}_1 \\ \ddot{x}_2 - u_2 - u_3 &= -\ddot{y}_2 \\ \ddot{\theta} + u_\alpha + u_\beta &= 0 \end{aligned} \qquad (16)$$

Eqs. (16) are applicable to the case where θ is not necessarily small.

2. Missile-Silo Isolation

Next we consider another problem which is quite similar to the one discussed in Ref. 2 but has a different arrange-

```
          2    MULTI-DEGREE-OF-FREEDOM SYSTEM
          2    0         3    3    1    1    41   0.15         2    1    0    1
              1
     M MATRIX
          1    1    1.
          2    2    1.
     1    3    3    1.
     U MATRIX
          1    1   -1.
          2    2   -1.
     1    3    3   -1.
     F MATRIX
     1    1    1   -1.0
     PX2 MATRIX
     0    1    1    1.0
     1    2    2    1.0
     Y1 MATRIX
     1    1    4    1.0
     FORCING FUNCTION
     1    1
     1    1    0
          1             1.612179
          2             0.941843
          3             0.457987
          4             0.117346
          5            -0.114306
          6            -0.263925
          7            -0.352623
          8            -0.396840
          9            -0.409288
         10            -0.399715
         11            -0.375526
         12            -0.342280
         13            -0.304089
         14            -0.263940
         15            -0.223950
         16            -0.185566
         17            -0.149732
         18            -0.117012
         19            -0.087688
         20            -0.061842
         21            -0.03941
         22            -0.020233
         23            -0.004090
         24             0.009276
         25             0.020144
         26             0.028789
         27             0.035481
         28             0.040480
         29             0.044026
         30             0.046339
         31             0.047618
         32             0.048042
         33             0.047769
         34             0.046936
         35             0.045664
```

Fig. 8. Input Deck for Multi-Degree-of-Freedom System

```
                36          0.044055
                37          0.042198
                38          0.040166
                39          0.038021
          1     40          0.035815
          BOUNDS OF CONSTRAINTS
          1     1     0           0
          0
              0.1
          0
             -0.1
          Q2 MATRIX
          1     1     1     1.0
          VERIFY
             1     1
          FINISH INPUT DATA
          STOP
```

Fig. 8. Input Deck for Multi-degree-of-Freedom System (concluded)

VEC. NO.	TIME	V(1)	V(2)	V(3)	V(4)
1	0.	2.948257E-01	0.	0.	-2.948257E-01
2	1.500000E-01	2.948257E-01	0.	0.	-2.948257E-01
3	3.000000E-01	2.948257E-01	0.	0.	-2.948257E-01
4	4.500000E-01	2.948257E-01	0.	0.	-2.948257E-01
5	6.000000E-01	2.948257E-01	0.	0.	-2.948257E-01
6	7.500000E-01	0.	0.	0.	0.
7	9.000000E-01	-2.948257E-01	0.	0.	2.948257E-01
8	1.050000E+00	-1.727954E-01	0.	0.	1.727954E-01
9	1.200000E+00	-2.948257E-01	0.	0.	2.948257E-01
10	1.350000E+00	0.	0.	0.	0.
11	1.500000E+00	-2.948257E-01	0.	0.	2.948257E-01
12	1.650000E+00	-2.948257E-01	0.	0.	2.948257E-01
13	1.800000E+00	0.	0.	[illegible]	0.
14	1.950000E+00	-2.246069E-01	0.	0.	[illegible]
15	2.100000E+00	0.	0.	0.	0.
16	2.250000E+00	-7.838257E-02	0.	0.	7.838257E-02
17	2.400000E+00	0.	0.	0.	0.
18	2.550000E+00	0.	0.	0.	0.
19	2.700000E+00	0.	0.	0.	0.
20	2.850000E+00	0.	0.	0.	0.
21	3.000000E+00	0.	0.	0.	0.
22	3.150000E+00	0.	0.	0.	0.
23	3.300000E+00	0.	0.	0.	0.
24	3.450000E+00	0.	0.	0.	0.
25	3.600000E+00	0.	0.	0.	0.
26	3.750000E+00	0.	0.	0.	0.
27	3.900000E+00	0.	0.	0.	0.
28	4.050000E+00	0.	0.	0.	0.
29	4.200000E+00	0.	0.	0.	0.
30	4.350000E+00	0.	0.	0.	0.
31	4.500000E+00	0.	0.	0.	0.
32	4.650000E+00	0.	0.	0.	0.
33	4.800000E+00	0.	0.	0.	0.
34	4.950000E+00	0.	0.	0.	0.
35	5.100000E+00	0.	0.	0.	0.
36	5.250000E+00	0.	0.	0.	0.
37	5.400000E+00	0.	0.	0.	0.
38	5.550000E+00	0.	0.	0.	0.
39	5.700000E+00	0.	0.	0.	0.
40	5.850000E+00	0.	0.	0.	0.

MAXIMUM VALUE OF CRITERION FUNCTION = 5.896513E-01

Table 1. Results of Multi-Degree-of-Freedom System given in Ref. 2.

TU= O1*S+ O2*U+ O3*F

WHERE

O1= 0.00 0.00 0.00 0.00 0.00 0.00
O2= 1.00 0.00 0.00
O3= 0.00 0.00

TIME INTERVAL			TU
0.000	TO	.150	.589653
.150	TO	.300	.589650
.300	TO	.450	.589650
.450	TO	.600	.589650
.600	TO	.750	.589650
.750	TO	.900	.117460
.900	TO	1.050	-.589650
1.050	TO	1.200	-.589650
1.200	TO	1.350	-.589650
1.350	TO	1.500	0.000000
1.500	TO	1.650	0.000000
1.650	TO	1.800	-.589650
1.800	TO	1.950	0.000000
1.950	TO	2.100	0.000000
2.100	TO	2.250	-.176940
2.250	TO	2.400	0.000000
2.400	TO	2.550	0.000000
2.550	TO	2.700	0.000000
2.700	TO	2.850	0.000000
2.850	TO	3.000	-.589650
3.000	TO	3.150	-.589650
3.150	TO	3.300	-.589650
3.300	TO	3.450	-.159170
3.450	TO	3.600	.589650
3.600	TO	3.750	0.000000
3.750	TO	3.900	0.000000
3.900	TO	4.050	0.000000
4.050	TO	4.200	0.000000
4.200	TO	4.350	0.000000
4.350	TO	4.500	0.000000
4.500	TO	4.650	.148460
4.650	TO	4.800	0.000000
4.800	TO	4.950	0.000000
4.950	TO	5.100	0.000000
5.100	TO	5.250	0.000000
5.250	TO	5.400	.311480
5.400	TO	5.550	0.000000
5.550	TO	5.700	0.000000
5.700	TO	5.850	.008950
5.850	TO	6.000	0.000000

OBJECTIVE FUNCTION = .589653 BOUND = .100000

Table 2.a Results for Multi-Degree-of-Freedom System

i	$u_1(t_i)$	$u_4(t_i)$	$w(t_i) = u_1 - u_4$	$u(t_i)$
1	0.2948257	-0.2948257	0.5896514	0.589653
2.	0.2848257	-0.2948257	0.5896514	0.583650
3.	0.2948257	-0.2948257	0.5896514	0.589650
4.	0.2948257	-0.2948257	0.5896514	0.589650
5.	0.2948257	-0.2948257	0.5896514	0.589650
6.	0.0	0.0	0.0	0.117460
7.	-0.2948257	0.2948257	-0.5896514	-0.589650
8.	-0.172954	0.1727954	-0.34559080	-0.589650
9.	-0.2948257	0.2948257	-0.5896514	-0.589650
10.	0.0	0.0	0.0	0.0

Table 2b. Comparison of Control Forces

Results obtained by Ref. 2 and those by PERFORM. u_1 and u_4 are taken from Ref. 2. u's were computed by PERFORM.

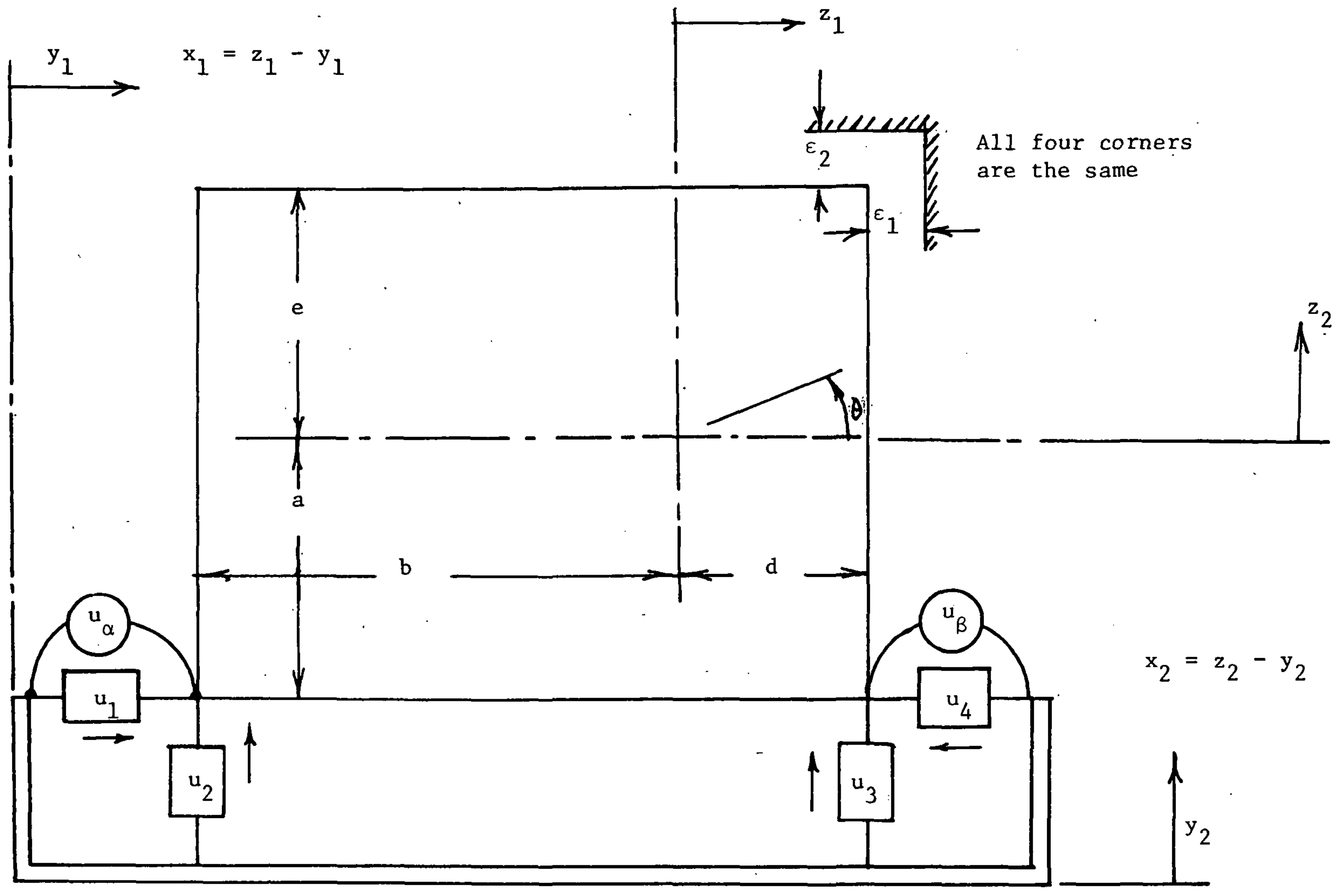


Fig.9. Modified Active Isolation System

ment of control forces. It is a problem of suspending a structure within an underground cavity so as to provide optimum shock isolation. We consider the structure to be a missile suspended in a silo in the manner shown by Fig. 10, where u_1, u_2, u_3, u_4, and u_5 represent shock isolators to be optimized. For purposes of analysis the missile is viewed as a rigid body capable of three modes of in-plane motion, two orthogonal translations and one rotation. The silo is assumed to be disturbed by a shock pulse, resulting in an in-plane motion of the silo enclosure. This motion can be decomposed into two orthogonal input motions, one vertical and one horizontal component. Consider Fig. 11, in which the notation is the same as that used in Fig. 7 except that β_i = angles of deviation of shock isolators from the original preshock directions.

The equations of motion are

$$
\begin{aligned}
&m\ddot{z}_1 + u_5 \sin \beta_5 + \sum_{i=1}^{4} u_i \cos \beta_i = 0 \\
&m\ddot{z}_2 + u_5 \cos \beta_5 + \sum_{i=1}^{4} u_i \sin \beta_i = 0 \\
&I\ddot{\theta} + u_5 h_2 \sin (\beta_5 + \theta) \\
&+ u_1 L_1 \left[\cos (\beta_1 - \theta) - \frac{d}{2L_1} \sin (\beta_1 - \theta)\right] \\
&- u_2 L_2 \left[\cos (\beta_2 - \theta) + \frac{d}{2L_2} \sin (\beta_2 - \theta)\right] \\
&+ u_3 L_1 \left[\cos (\beta_3 + \theta) - \frac{d}{2L_1} \sin (\beta_3 + \theta)\right] \\
&- u_4 L_2 \left[\cos (\beta_4 + \theta) + \frac{d}{2L_2} \sin (\beta_4 + \theta)\right] = 0
\end{aligned}
\tag{17}
$$

where m is the total mass of the missile and I is the moment of inertia of the missile. Since β_i are functions of z_1, z_2, y_1, y_2, the terms such as $\beta_i u_i$ in the above equations are nonlinear.

Now assume, as in the previous section, that the isolator forces retain their horizontal or vertical directions despite the motions of the base and mass. Again, assume the θ is small, and in addition assume β_i is small. Then the equations of motion are rewritten as

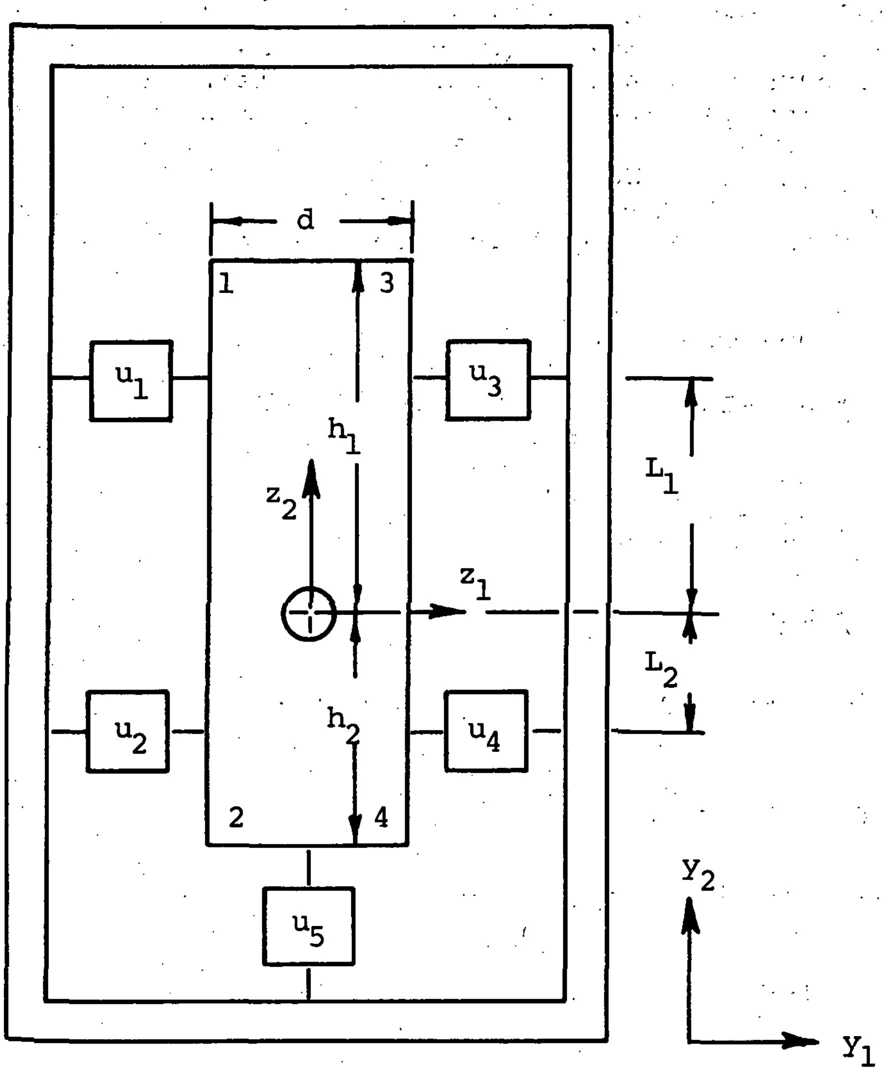


Fig 10. Model Representing the In-Plane Three-Degree-of Freedom Silo-Missile System

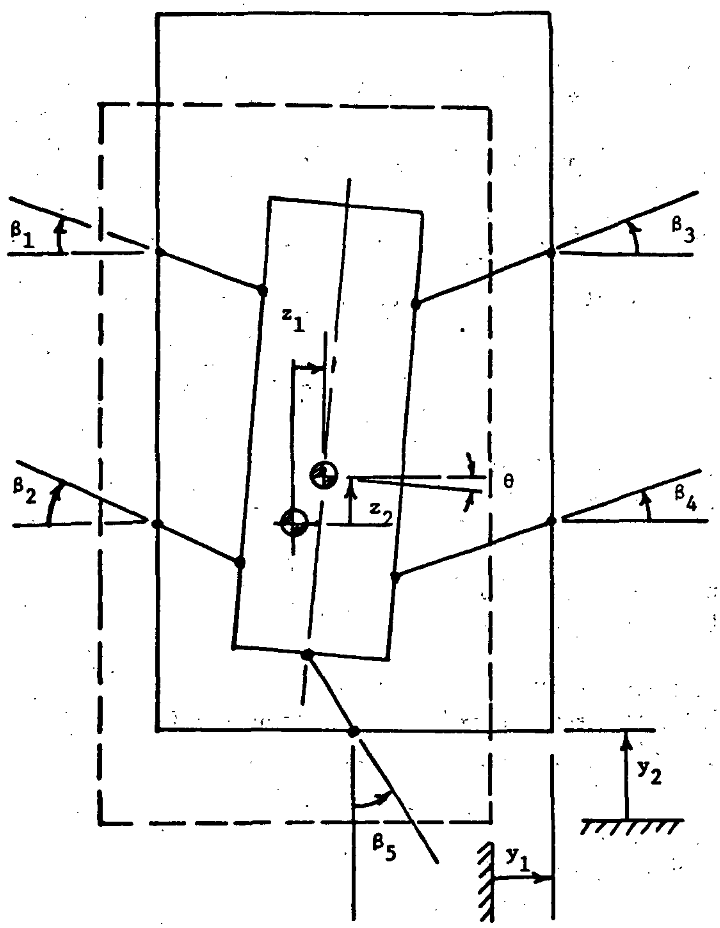


Fig. 11 Kinematics of the Three-Degree-of-Freedom Model

$$m\ddot{z}_1 + \sum_{i=1}^{4} u_i = 0$$

$$m\ddot{z}_2 + u_5 = 0 \tag{18}$$

$$I\ddot{\theta} + u_1L_1 - u_2L_2 + u_3L_1 - u_4L_2 = 0$$

The relations

$$z_1 = x_1 + y_1,\ \dot{z}_1 = \dot{x}_1 + \dot{y}_1,\ \ddot{z}_1 = \ddot{x}_1 + \ddot{y}_1$$

$$z_2 = x_2 + y_2,\ \dot{z}_2 = \dot{x}_2 + \dot{y}_2,\ \ddot{z}_2 = \ddot{x}_2 + \ddot{y}_2$$

and $I = m\rho^2$ substituted in Eq. (16) yield, for unit mass,

$$\ddot{x}_1 + u_1 + u_2 + u_3 + u_4 = -\ddot{y}_1$$

$$\ddot{x}_2 + u_5 = -\ddot{y}_2 \tag{19}$$

$$\ddot{\theta} + u_1\frac{L_1}{\rho^2} - u_2\frac{L_2}{\rho^2} + u_3\frac{L_1}{\rho^2} - u_4\frac{L_2}{\rho^2} = 0$$

The above equations are all linear and in the desired form for PERFORM input format.

With additional control forces $u_{\alpha 1}$, $u_{\alpha 2}$, $u_{\alpha 3}$, and $u_{\alpha 4}$ shown in Fig. 12 and neglecting the contribution of u_1, u_2, u_3, and u_4 to rotational motion, the equations of motion can be written as

$$\ddot{x}_1 + u_1 + u_2 + u_3 + u_4 = -\ddot{y}_1$$

$$\ddot{y}_2 + u_5 = -\ddot{y}_2 \tag{20}$$

$$\ddot{\theta} + \frac{u_{\alpha 1}}{\rho^2} + \frac{u_{\alpha 2}}{\rho^2} + \frac{u_{\alpha 3}}{\rho^2} + \frac{u_{\alpha 4}}{\rho^2} = 0$$

Again we have equations of motion in the desired form for the PERFORM input format and the equations are applicable where θ and β_i are not necessarily small.

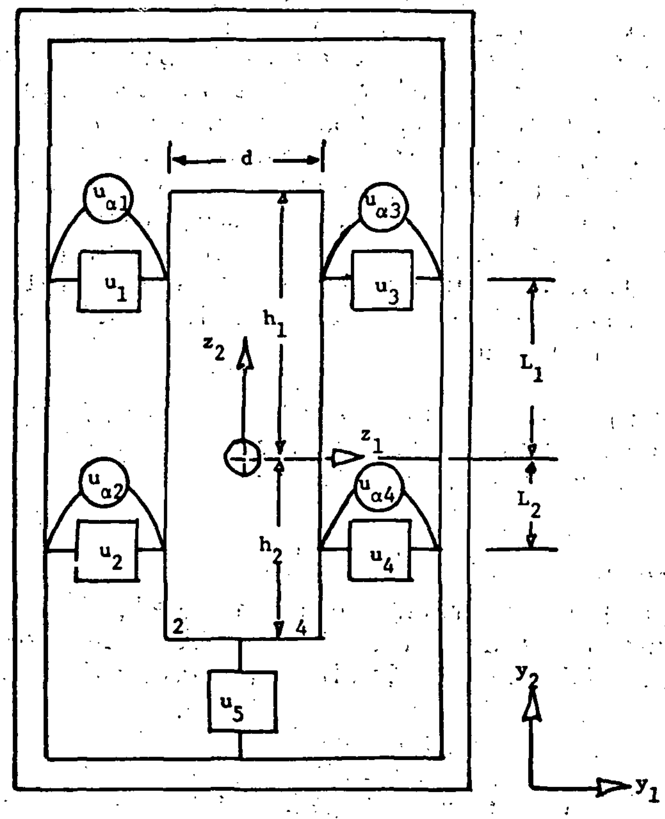


Fig. 12 Modified Silo-Missle System

C. Train Suspension Model

The proper design of the wheel-to-car (or truck) suspension mechanism is essential to the successful operation of a high speed rail passenger car. Simplified vertical and lateral dynamic models of a high speed railroad car have been developed and a trial-and-error optimization procedure has been applied to determine the optimum stiffness and damping properties of the suspension systems in Ref. 3.

The mathematical models used were a four-degree-of-freedom model for vertical responses to vertical inputs and a ten-degree-of-freedom model for lateral responses to lateral or rolling inputs from the rails. The linear equations of motion were solved by digital computer programs for sinusoidal inputs to both models and for one random input to the vertical model. To include nonlinearities in the lateral truck suspension system, a real-time digital simulation computer program was developed and utilized. Both linear and nonlinear acceleration responses to sinusoidal, transient deterministic, and random inputs were obtained for the lateral model by means of this simulation program.

In this work we formulate the application of the computer system PERFORM in evaluating the limiting performance of the lateral train model, subject to transient inputs.

1. Description of the System

The lateral train model used in Ref. 3 is a ten degree-of-freedom system as shown in Fig. 13. The motion of this model is described by the following generalized coordinates:

$b(t)$ - car lateral bending

$y_c(t)$ - car lateral rigid-body translation

$\theta_{co}(t)$ - car rigid body roll

$\theta_{c1}(t)$ - car torsion

$\eta(t)$ - car rigid-body yaw

$y_{\ell g}(t)$ - transformer lateral translation

$y_{Ar}(t)$ - rear traction motor lateral translation

$y_{Af}(t)$ - forward traction motor lateral translation

$\theta_{Ar}(t)$ - rear traction motor roll

$\theta_{Af}(t)$ - forward traction motor roll

For a limiting performance study we replace the spring and damping forces in Fig. 13 by the control or isolator forces as shown in Figs. 14 and 15. Fig. 14 represents a general case where the entire spring and damping forces of the system are replaced by eleven isolator forces. A special case of only 3 isolator forces is shown in Fig. 15. The eleven isolator forces considered are:

u_{cr} - rear lateral bolster force

u_{cf} - forward lateral bolster force

u_{sr} - rear vertical bolster force

u_{sf} - forward vertical bolster force

u_{g} - transformer lateral bolster force

$u_{\ell r}$ - rear traction motor rolling torque

$u_{\ell f}$ - forward traction motor rolling torque

u_{Ar} - rear vertical equalizer force

u_{Af} - forward vertical equalizer force

$u_{\alpha r}$ - rear traction motor rolling torque

$u_{\alpha f}$ - forward traction motor rolling torque

2. Operating Conditions and Objectives

The railroad car system is subjected to transient lateral and cross-level inputs which simulate the effects of a lateral deviation of the track from a straight course and the bank on a curve, respectively.

The objective is to find the isolator forces that minimize a maximum lateral acceleration of the car. Furthermore, the constraints are on the lateral deflection of the bolster springs. Thus we wish to find the unknown isolator forces such that

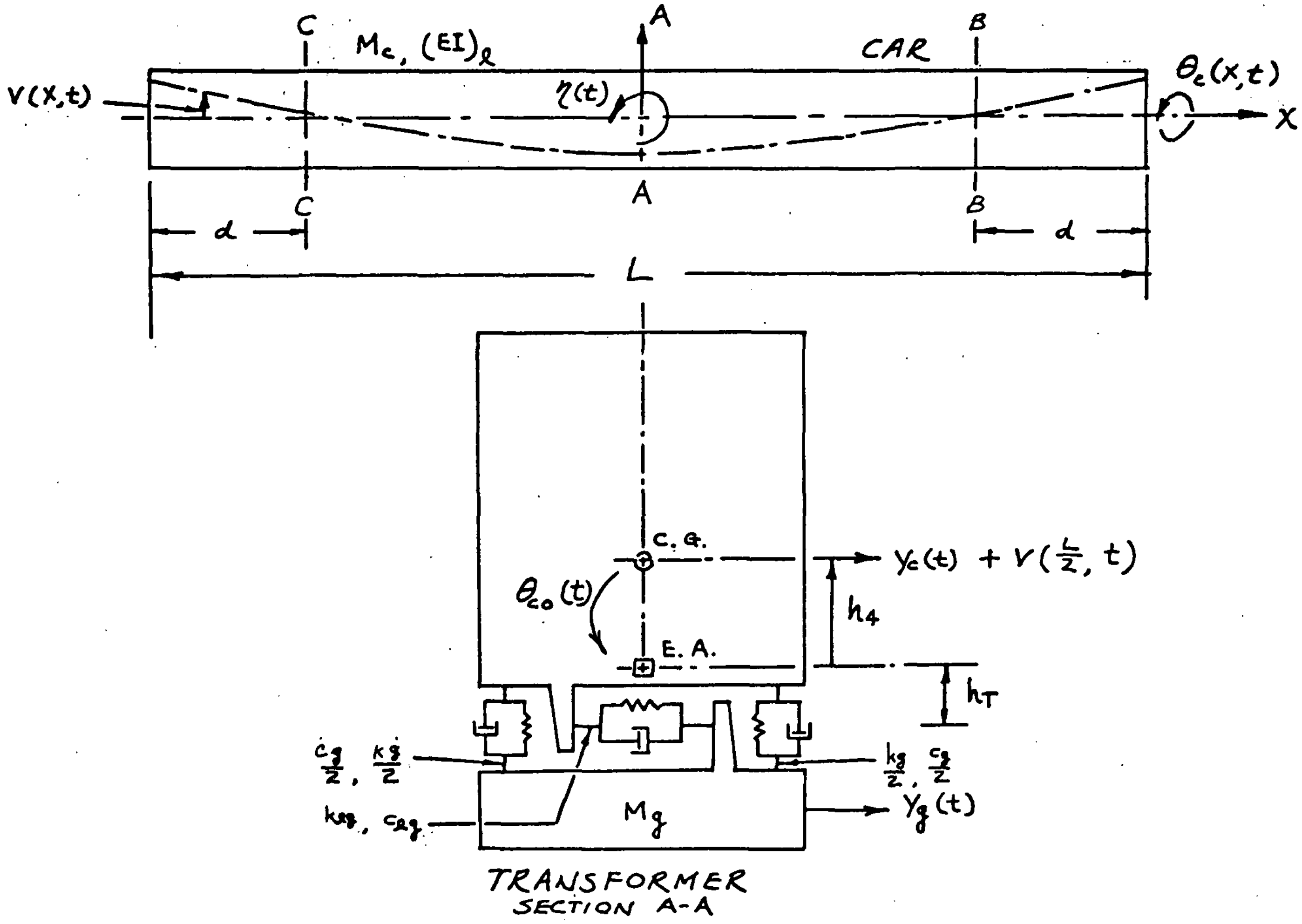


Fig. 13 Mathematical Lateral Model of Railroad Car and Truck Suspension

(a) Top and Mid Section Views

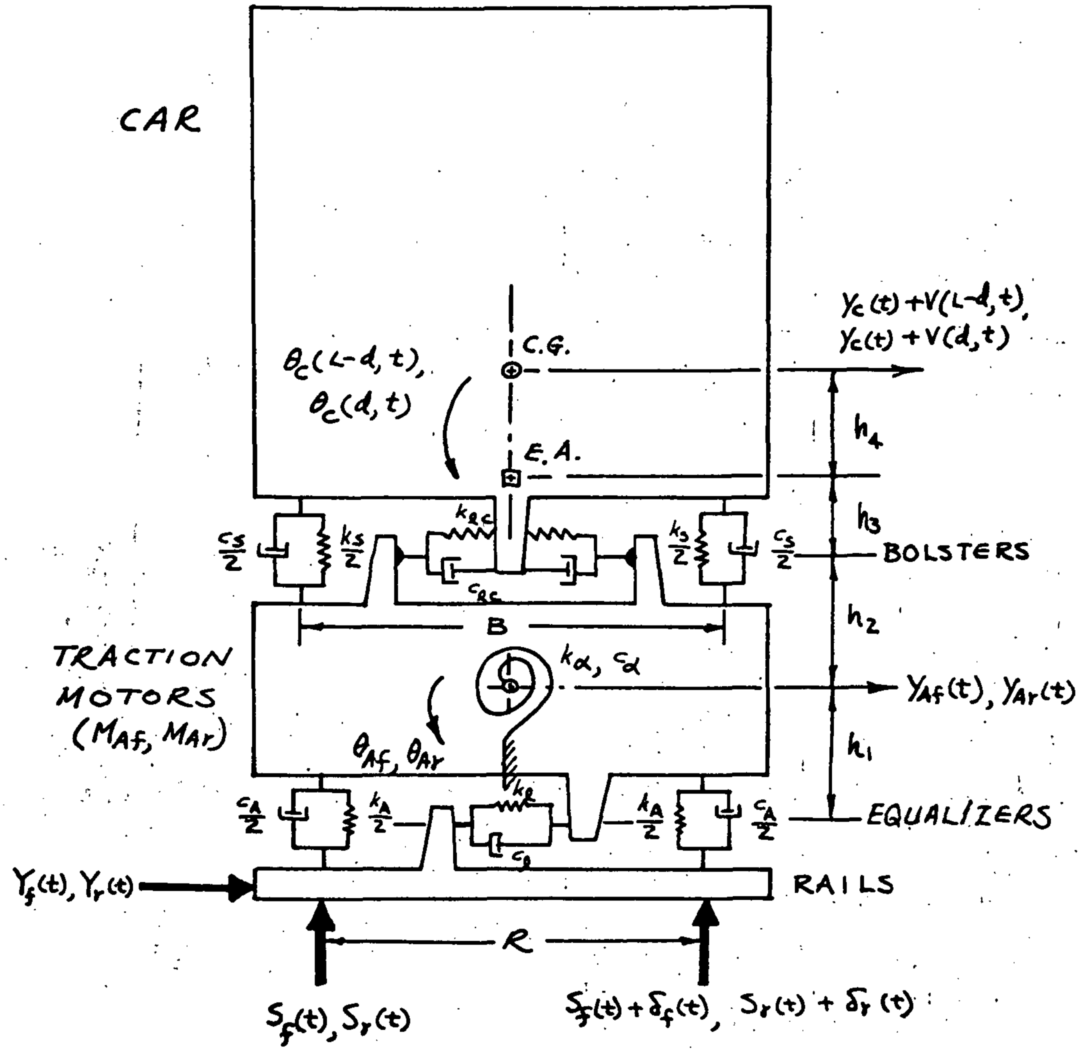


Fig. 13 Concluded

(b) Truck Suspension (Section B-B, and C-C)

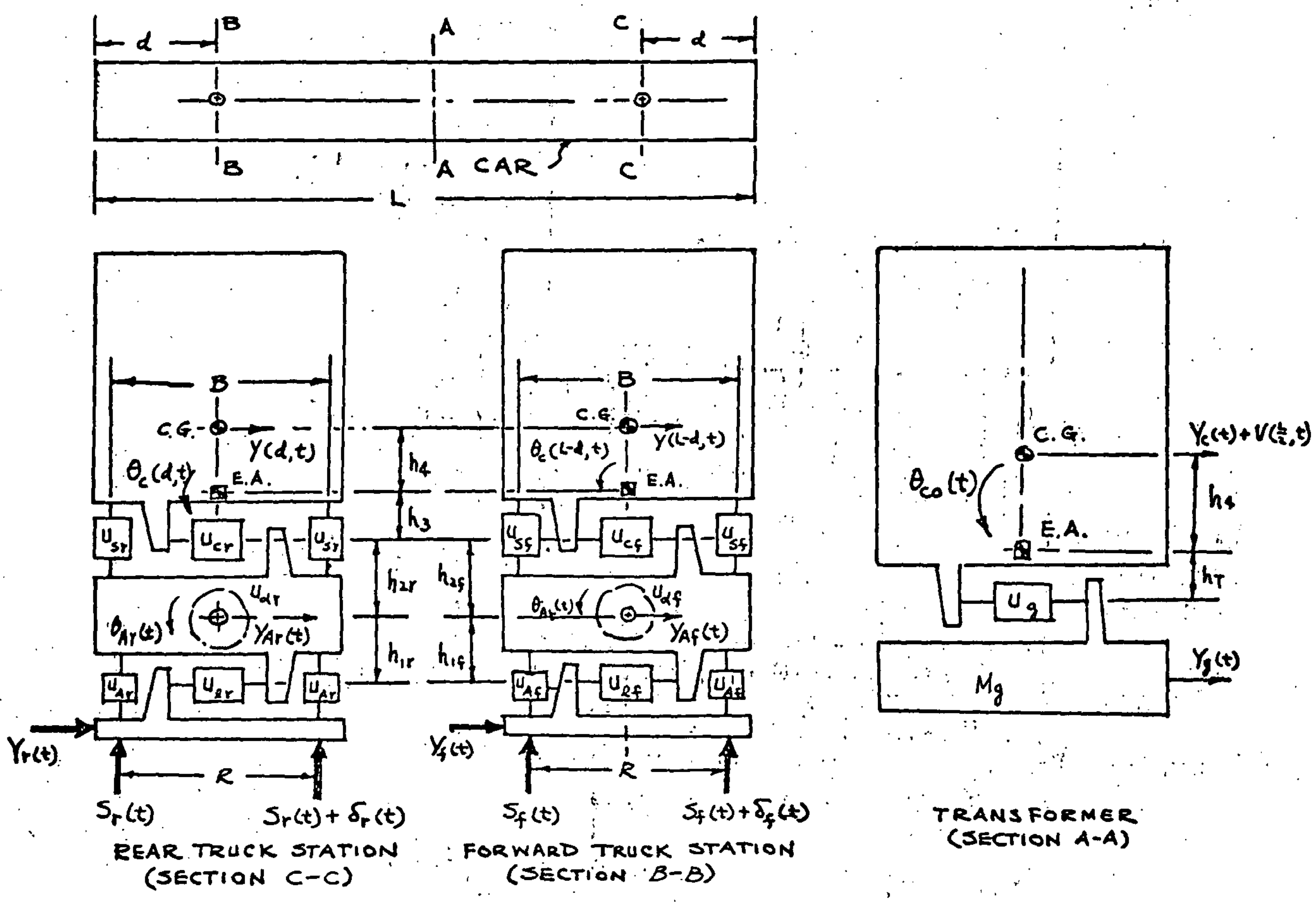


Fig. 14 Schematic View of Lateral Rail Car Model
With Eleven Isolator Forces

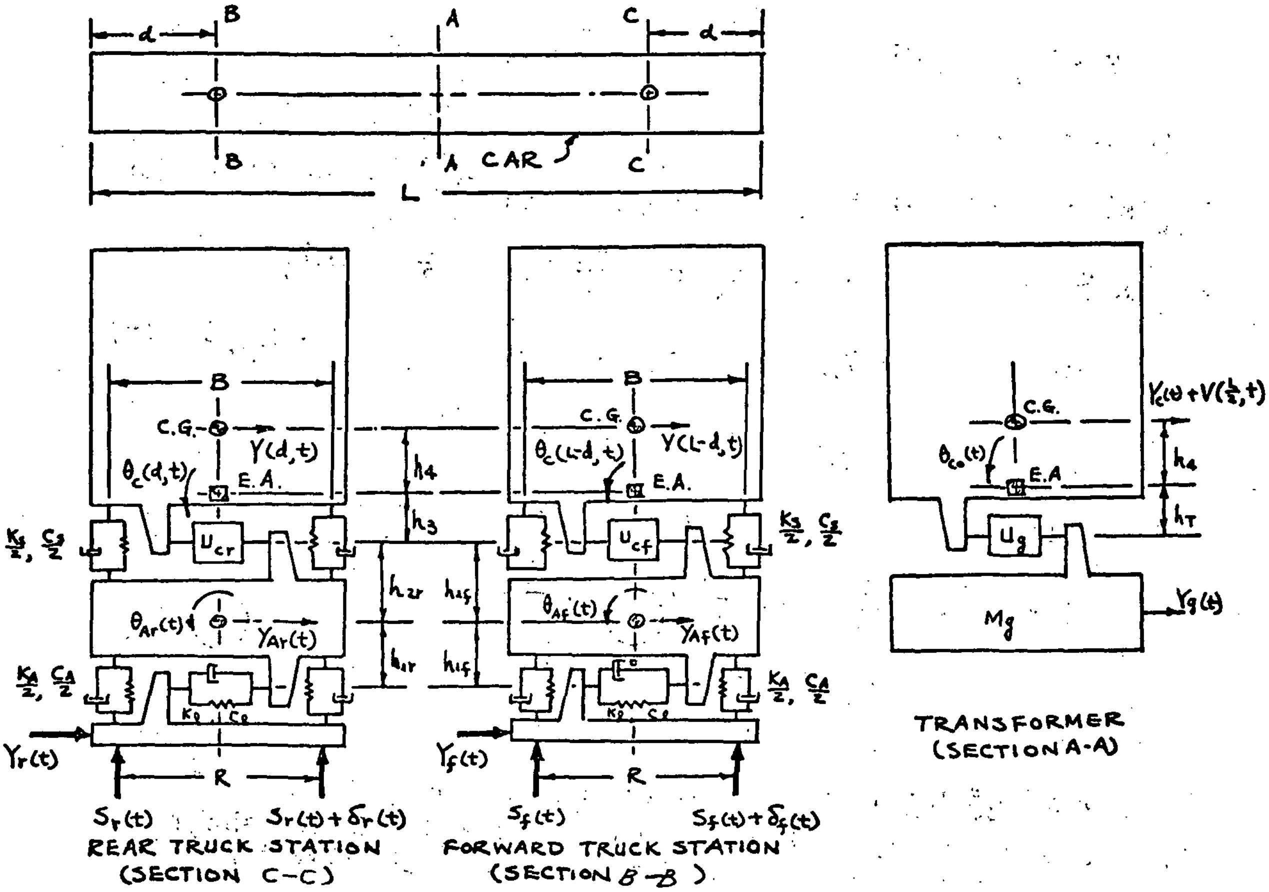


Fig. 15 Schematic View of Lateral Rail Car Model
With Three Isolator Forces

$$\max |A_{c\ell}(x)|$$

is minimized, and

$$YL \le \Delta Y_r(t) \le YU$$

$$YL \le \Delta Y_f(t) \le YU$$

are satisfied. Here

$$A_{c\ell}(x) = \ddot{b}(t)W(x) + \ddot{Y}_c - h_4(\ddot{\Theta}_{co} + \ddot{\Theta}_{c1} \cos \frac{\pi x}{L}) - (\frac{L}{2} - x)\ddot{\eta}$$

$$= W(x)\ddot{q}_1 + \ddot{q}_2 - h_4\ddot{q}_3 - (h_4 \cos \frac{\pi x}{L})\ddot{q}_4 - (\frac{L}{2} - x)\ddot{q}_5$$

$$\Delta Y_r = -b(t)W(d) - Y_c(t) + (\frac{L}{2} - d)\eta(t) + Y_{Ar}(t) - h_{2r}\Theta_{Ar}(t)$$
$$-h_3[\Theta_{co}(t) + \Theta_{c1}(t) \cos \frac{\pi d}{L}]$$

ΔY_f = same as ΔY_r with $W(d)$ replaced by $W(L-d)$,

$(\frac{L}{2} - d)$ by $-(\frac{L}{2} - d)$, $\frac{\pi d}{L}$ by $\frac{\pi(L-d)}{L}$

and the subscript r by f

3. Modified Equations of Motion for PERFORM

The equations of motion for the original system shown in Fig. 13 were derived in Ref. 3 and expressed in the form

$$\underline{M}\ddot{\bar{q}} + \underline{C}\dot{\bar{q}} \pm \underline{K}\bar{q} = \underline{F}\bar{f}_k \tag{21}$$

In order to use PERFORM the equations of motion must appear as

$$\underline{M}\,\ddot{\bar{q}} + \underline{C}\,\dot{\bar{q}} + \underline{K}\,\bar{q} + \underline{U}\,\bar{u} = \underline{F}\,\bar{f}_k \tag{22}$$

where the u's are the control forces that replace the spring and damping forces.

Formulation of equations of motion in the form of Eq. (22) is not unique and depends on the choice of the coordinate system as well as the isolator forces to be used. This will be shown in the following illustrations with a two-degree-of-freedom system with its kinematic relations as indicated in Fig. 16a. The equations of motion in the coordinates z_1 and z_2 are written as

$$\begin{aligned} m_1\ddot{z}_1 + k_1(z_1-f) + k_2(z_1-z_2) &= 0 \\ m_2\ddot{z}_2 + k_2(z_2-z_1) &= 0 \end{aligned} \tag{23}$$

Using the coordinates x_1 and x_2 the equations of motion become

$$\begin{aligned} m_1\ddot{x}_1 + k_1x_1 - k_2\,x_2 &= -m_1\ddot{f} \\ m_2(\ddot{x}_2+\ddot{x}_1) + k_2\,x_2 &= -m_2\ddot{f} \end{aligned} \tag{24}$$

Now we write the equations of motion in the form of Eqs. (23) and (24) for the following six different cases of coordinate systems and isolator forces.

Case 1 (Fig. 16b)

$$\begin{aligned} m_1\ddot{z}_1 + u_1 + u_2 &= 0 \\ m_2\ddot{z}_2 - u_2 &= 0 \end{aligned} \tag{25}$$

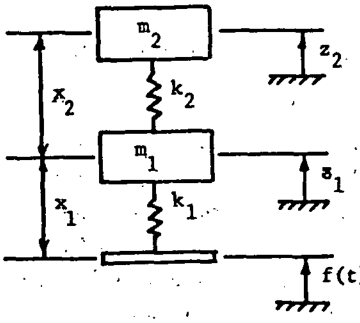
where

$$u_1 = k_1\ (z_1-f)$$

$$u_2 = k_2(z_1-z_2)$$

Case 2 (Fig. 16c)

$$m_1\ddot{x}_1 + u_1 + u_2 = -m_1\ddot{f}$$



$$z_1 = x_1 + f$$

$$z_2 = x_1 + x_2 + f$$

Fig. (a)

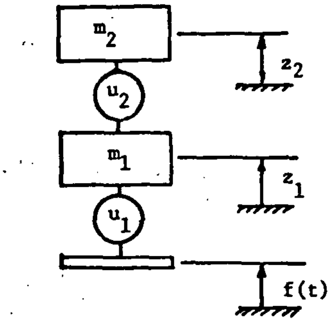


Fig. (b)

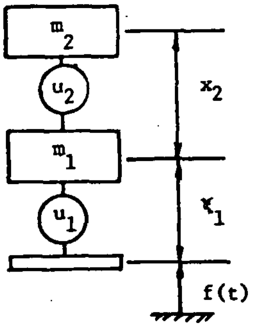


Fig. (c)

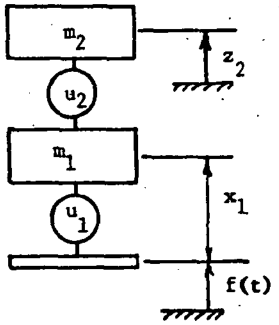


Fig. (d)

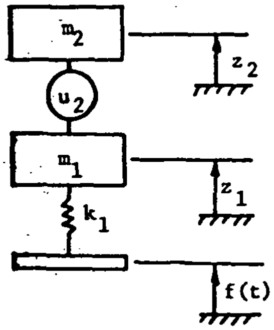


Fig. (e)

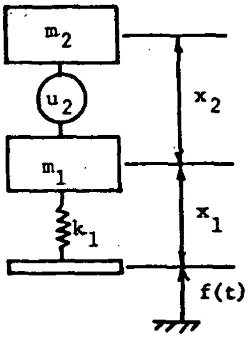


Fig. (f)

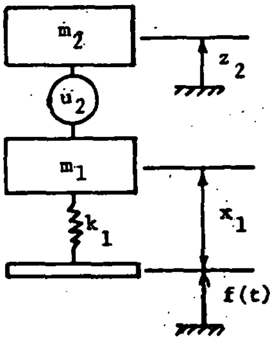


Fig. (g)

Fig.16 Two-Degree-of-Freedom System

$$m_2(\ddot{x}_2 + \ddot{x}_1) - u_2 = -m_2\ddot{f} \tag{26}$$

where

$$u_1 = k_1 x_1$$

$$u_2 = -k_2 x_2$$

Case 3 (Fig. 16d)

$$\begin{aligned} m_1\ddot{x}_1 + u_1 + u_2 &= -m_1\ddot{f} \\ m_2\ddot{z}_2 - u_2 &= 0 \end{aligned} \tag{27}$$

Case 4 (Fig. 16e)

$$\begin{aligned} m_1\ddot{z}_1 + k_1 z_1 + u_2 &= k_1 f \\ m_2\ddot{z}_2 - u_2 &= 0 \end{aligned} \tag{28}$$

where

$$u_2 = k_2(z_1 - z_2)$$

Case 5 (Fig. 16f)

$$\begin{aligned} m_1\ddot{x}_1 + k_1 x_1 + u_2 &= -m_1\ddot{f} \\ m_2(\ddot{x}_1 + \ddot{x}_2) - u_2 &= -m_2\ddot{f} \end{aligned} \tag{29}$$

where

$$u_2 = -k_2 x_2$$

Case 6 (Fig. 16g)

$$\begin{aligned} m_1\ddot{x}_1 + k_1 x_1 + u_2 &= -m_1\ddot{f} \\ m_2\ddot{z}_2 - u_2 &= 0 \end{aligned} \tag{30}$$

In Case 1, where the coordinates z_1 and z_2 represent absolute displacements of masses m_1 and m_2, the equations of motion (Eq. (25)) do not have any forcing functions. This would not be an acceptable form unless only initial conditions of these coordinates are specified. Furthermore, if for Case 1 we

define

$$u_1 = k_1 z_1 \text{ and}$$

$$u_2 = k_2 (z_1 - z_2),$$

the equations of motion become

$$\begin{aligned} m_1 \ddot{z}_1 + u_1 + u_2 &= k_1 f \\ m_2 \ddot{z}_2 - u_2 &= 0 \end{aligned} \tag{31}$$

Eq. (31) includes a forcing function and may appear to be as acceptable as Case 4. However, we have here the problem of a term with the system spring characteristics with k_1 not being completely replaced by u_1 or u_2. Cases 2 through 6 are all acceptable and the choice of a particular coordinate system or isolator forces depends on the initial conditions and the nature of the problem at hand.

a. The Case of Eleven Isolator Forces (Fig. 14).

For this case the equations of motion in the form of Eq. (22) will have the following vectors and coefficient matrices:

. Displacement Vector

$$\bar{q} = \begin{bmatrix} b = q_1 \\ y_c = q_2 \\ \Theta_{co} = q_3 \\ \Theta_{c1} = q_4 \\ \eta = q_5 \\ y_{1g} = q_6 \\ z_r = q_7 \\ z_f = q_8 \\ \phi_r = q_9 \\ \phi_f = q_{10} \end{bmatrix}$$

where

$$z_r = y_{Ar} - Y_r$$

$$z_f = y_{Af} - Y_f$$

$$\phi_r = \theta_{Ar} - \frac{\delta_r}{R}$$

$$\phi_f = \theta_{Af} - \frac{\delta_f}{R}$$

The coordinate system used here is similar to that of Case 3 discussed above.

. Forcing Function Vector

$$\bar{f}_1 = \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{bmatrix}$$

where

$$f_{11} = M_{Ar}\,\ddot{Y}_r$$

$$f_{12} = M_{Af}\,\ddot{Y}_f$$

$$f_{13} = I_{Ar}\,\ddot{\delta}_r$$

$$f_{14} = I_{Af}\,\ddot{\delta}_f$$

and

$$Y_f = \frac{Y_{of}}{2}\left[1 - \mathrm{Cos}\,\frac{2\pi V(t-t_o)}{\lambda + (\Delta\ell)_t}\right], \qquad \ddot{Y}_f = \frac{2\,Y_{of}\,(\pi V)^2}{[\lambda + (\Delta\ell)_t]^2}\ \mathrm{Cos}\,\frac{2\pi V(t-t_o)}{\lambda + (\Delta\ell)_t}$$

$$Y_r = \frac{Y_{or}}{2}\left[1 - \mathrm{Cos}\,\frac{2\pi V(t-t_2)}{\lambda + (\Delta\ell)_t}\right], \qquad \ddot{Y}_r = \frac{2\,Y_{or}\,(\pi V)^2}{[\lambda + (\Delta\ell)_t]^2}\ \mathrm{Cos}\,\frac{2\pi V(t-t_2)}{\lambda + (\Delta\ell)_t}$$

$$\delta_r = \frac{\delta_{or}}{2}\left[1 - \text{Cos}\,\frac{2\pi V(t-t_o)}{\lambda + (\Delta\ell)_t}\right], \quad \ddot{\delta}_r = \frac{2\,\delta_{or}\,(\pi V)^2}{[\lambda + (\Delta\ell)_t]^2}\,\text{Cos}\,\frac{2\pi V(t-t_o)}{\lambda + (\Delta\ell)_t}$$

$$\delta_f = \frac{\delta_{of}}{2}\left[1 - \text{Cos}\,\frac{2\pi V\,(t-t_2)}{\lambda + (\Delta\ell)_t}\right], \quad \ddot{\delta}_f = \frac{2\,\delta_{of}\,(\pi V)^2}{[\lambda + (\Delta\ell)_t]^2}\,\text{Cos}\,\frac{2\pi V(t-t_2)}{\lambda + (\Delta\ell)_t}$$

The Y's are lateral inputs and the δ's are the cross-level inputs. The second derivatives of Y's and δ's with respect to time are expressed as $\ddot{Y}$'s and $\ddot{\delta}$'s. The quantities V, λ, and $(\Delta\ell)_t$ represent the train speed, the wave length of the input disturbance, and the wheel base of a truck suspension, respectively. In the above equations t_2 is obtained from the relations

$$t_1 - t_o = \frac{\lambda + (\Delta\ell)_t}{V}$$

$$t_2 - t_1 = \frac{L - 2d - (\Delta\ell)_t - \lambda}{V}$$

where $t_1 - t_o$ = time interval for forward truck to roll over the disturbance.

$t_2 - t_1$ = time interval between when the front truck leaves the disturbance and the rear truck encounters it.

- Isolator Force Vector

$$\bar{u} = [u_{cr} = u_1,\ u_{cf} = u_2,\ u_{sr} = u_3,$$
$$u_{sf} = u_4,\ u_g = u_5,\ u_{\ell r} = u_6,$$
$$u_{\ell f} = u_7,\ u_{Ar} = u_8,\ u_{Af} = u_9,$$
$$u_{\alpha r} = u_{10},\ u_{\alpha f} = u_{11}].$$

where the isolator forces are defined in terms of the configuration of Fig. 13. They are

$$u_{cr} = u_1 = c_{\ell cr}\ [W(d)\dot{b} + \dot{y}_c + h_3\ \dot{\theta}_{co} + h_3 \cos\frac{\pi d}{L}\ \dot{\theta}_{cl} - (\frac{L}{2} - d)\dot{\eta}$$

$$-(\dot{z}_r + \dot{Y}_r) + h_{2r}\ (\dot{\phi}_r + \frac{\dot{\delta}_r}{R})] + k_{\ell cr}\ [W(d)b + y_c + h_3\ \theta_{co}$$

$$+ h_3 \cos\frac{\pi d}{L}\ \theta_{cl} - (\frac{L}{2} - d)\eta\ - (z_r + Y_r) + h_{2r}\ (\phi_r + \frac{\delta_r}{R})]$$

$u_{cf} = u_2$ Same as u_{cr} with the subscript r replaced by f, $W(d)$ by $W(L-d)$,

$-(\frac{L}{2} -d)$ by $(\frac{L}{2} -d)$, and $\frac{\pi d}{L}$ by $\frac{\pi(L-d)}{L}$

$$u_{sr} = u_3 = c_s\ [\frac{1}{2}\ \dot{\theta}_{co} + \frac{1}{4}\cos\frac{\pi d}{L}\ \dot{\theta}_{cl} - \frac{1}{4}\ (\dot{\phi}_r + \frac{\dot{\delta}_r}{R})]$$

$$+ k_s\ [\frac{1}{2}\ \theta_{co} + \frac{1}{4}\cos\frac{\pi d}{L}\ \theta_{cl} - \frac{1}{4}\ (\phi_r + \frac{\delta_r}{R})]$$

$u_{sf} = u_4$ = Same as u_{sr} with d replaced by (L-d) and the subscript r by f.

$$u_g = u_5 = c_{\ell g}\ [W(\frac{L}{2})\ \dot{b} + \dot{y}_c + h_T\ \dot{\theta}_{co} - \dot{Y}_g]$$

$$+ k_{\ell g}\ [W(\frac{L}{2})b + y_c + h_T\ \theta_{co} - Y_g]$$

$$u_{\ell r} = u_6 = c_{\ell r}\ [\dot{z}_r + h_{1r}\ (\dot{\phi}_r + \frac{\dot{\delta}_r}{R})] + k_{\ell r}\ [z_r + h_{1r}\ (\phi_r + \frac{\delta_r}{R})]$$

$u_{\ell f} = u_7$ = Same as $u_{\ell r}$ with subscript r replaced by f.

$$u_{Ar} = u_8 = c_{Ar}\ (\dot{\phi}_r + \frac{\dot{\delta}_r}{R}) + k_{Ar}\ (\phi_r + \frac{\delta_r}{R})$$

$u_{Af} = u_9$ = Same as u_{Ar} with subscript r replaced by f.

$$u_{\alpha r} = u_{10} = c_{\alpha r}\ (\dot{\phi}_n + \frac{\dot{\delta}_r}{R}) + k_{\alpha r}\ (\phi_r + \frac{\delta_r}{R})$$

$u_{\alpha f} = u_{11}$ = Same as $u_{\alpha r}$ with subscript r replaced by f.

. Damping Matrix

$\underline{C} = [0]$

- Spring Matrix

$\underline{K} = [0]$

except

$$K_{11} = (EI)_L \beta_1^4 L$$

$$K_{44} = \frac{GJL}{2} \left(\frac{\pi}{L}\right)^2$$

MASS MATRIX, $\underline{M}$

	1	2	3	4	5	6	7	8	9	10
1	M_c	0	0	$-mh_4\beta_1\{\}^{(a)}$	0	0	0	0	0	0
2	0	M_c	$-M_c h_4$	0	0	0	0	0	0	0
3	0	$-M_c h_4$	$I_{EA}L$	0	0	0	0	0	0	0
4	$-mh_4\beta_1\{\}^{(a)}$	0	0	$I_{EA}\frac{L}{2}$	$2mh_4\left(\frac{L}{\pi}\right)^2$	0	0	0	0	0
5	0	0	0	$2mh_4\left(\frac{L}{\pi}\right)^2$	$M_c\frac{L^2}{12}$	0	0	0	0	0
6	0	0	0	0	0	M_g	0	0	0	0
7	0	0	0	0	0	0	M_{Ar}	0	0	0
8	0	0	0	0	0	0	0	M_{Af}	0	0
9	0	0	0	0	0	0	0	0	I_{Ar}	0
10	0	0	0	0	0	0	0	0	0	I_{Af}

(a) $\{\} = -\left[\frac{\sin h\beta_1 L}{\beta_1^2 + (\frac{\pi}{L})^2} + \frac{\sin \beta_1 L}{\beta_1^2 - (\frac{\pi}{L})^2}\right] + \alpha_1 \left[\frac{\cos h\beta_1 L + 1}{\beta_1^2 + (\frac{\pi}{L})^2} - \frac{\cos \beta_1 L + 1}{\beta_1^2 - (\frac{\pi}{L})^2}\right]$; $\beta_1 L = 4.730$, $\alpha_1 = 0.9825$

. *Control Force Matrix*

$$
\underline{U} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array}
\begin{array}{c}
\begin{array}{ccccccccccc} (1) & (2) & (3) & (4) & (5) & (6) & (7) & (8) & (9) & (10) & (11) \end{array} \\
\left[\begin{array}{ccccccccccc}
W(d) & W(L-d) & 0 & 0. & W(\frac{L}{2}) & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
h_3 & h_3 & B^2 & B^2 & h_T & 0 & 0 & 0 & 0 & 0 & 0 \\
h_3 \cos\frac{\pi d}{L} & h_3 \cos\frac{\pi(L-d)}{L} & B^2 \cos\frac{\pi d}{L} & B^2 \cos\frac{\pi(L-d)}{L} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-(\frac{L}{2} - d) & (\frac{L}{2} - d) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
h_{2r} & 0 & -B & 0 & 0 & h_{r1} & 0 & \frac{R^2}{4} & 0 & 1 & 0 \\
0 & h_{2f} & 0 & -B & 0 & 0 & h_{1f} & 0 & \frac{R^2}{4} & 0 & 1
\end{array}\right]
\end{array}
$$

• Forcing Function Matrix

$$\underline{F} = \begin{array}{c} \\ (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \\ (7) \\ (8) \\ (9) \\ (10) \end{array} \begin{array}{c} \begin{array}{cccc} (1) & (2) & (3) & (4) \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -\frac{1}{R} & 0 \\ 0 & 0 & 0 & -\frac{1}{R} \end{bmatrix} \end{array}$$

b. The Case of Three Isolator Forces (Fig. 15)

The vectors and coefficient matrices of the equations of motion in the form of Eq. (22) are defined as follows:

. Displacement Vector

$$\bar{q} = \begin{bmatrix} b = q_1 \\ y_c = q_2 \\ \theta_{co} = q_3 \\ \theta_{cl} = q_4 \\ \eta = q_5 \\ y_g = q_6 \\ y_{Ar} = q_7 \\ y_{Af} = q_8 \\ \theta_{Ar} = q_9 \\ \theta_{Af} = q_{10} \end{bmatrix}$$

Here we are using a coordinate system similar to that of Case 4.

. Forcing Function Vector

$$\bar{f}_1 = \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \end{bmatrix}$$

where

$$f_{11} = k_{\ell r}\, Y_r + c_{\ell r}\, \dot{Y}_r$$

$$f_{12} = k_{\ell f}\, Y_f + c_{\ell f}\, \dot{Y}_f$$

$$f_{13} = k_{Ar}\, \delta_r + c_{Ar}\, \dot{\delta}_r$$

$$f_{14} = k_{Af}\, \delta_f + c_{Af}\, \dot{\delta}_f$$

The rail inputs Y's and δ's are the same as in the first case. $\dot{Y}$'s and $\dot{\delta}$'s are the time derivatives

$$\dot{Y}_r = \frac{Y_{or}\ \pi V}{+\ (\Delta\ell)_t} \quad \sin \frac{2\pi V\ (t-t_2)}{\lambda + (\Delta\ell)_t}$$

$$\dot{Y}_f = \frac{Y_{of}\ \pi V}{\lambda + (\Delta\ell)_t} \quad \sin \frac{2\pi V\ (t-t_o)}{\lambda + (\Delta\ell)_t}$$

The expressions for $\dot{\delta}_r$ and $\dot{\delta}_f$ are of the same form as for the $\dot{Y}_r$ and $\dot{Y}_f$ with δ_{of} and δ_{or} replacing Y_{of} and Y_{or}.

. Control Force Vector

$$\bar{u} = \begin{bmatrix} u_{cr} = u_1 \\ u_{cf} = u_2 \\ u_g = u_3 \end{bmatrix}$$

where the isolator forces are defined in terms of the configuration of Fig. 13. These are

$$u_1 = c_{\ell cr}\ [W(d)\dot{b} + \dot{y}_c + h_3\ \dot{\theta}_{co} + h_3 \cos \frac{\pi d}{L}\ \dot{\theta}_{cl} - (\frac{L}{2} - d)\dot{\eta}$$

$$- \dot{y}_{Ar} + h_{2r}\ \dot{\theta}_{Ar}]$$

$$+ k_{\ell cr}\ [W(d)b + y_c + h_3\ \theta_{co} + h_3 \cos \frac{\pi d}{L}\ \theta_{cl} - (\frac{L}{2} - d)\eta$$

$$- y_{Ar} + h_{2r}\ \theta_{Ar}]$$

u_2 = Same as u_1 with the subscript r replaced by f, and W(d) by W(L-d), $-(\frac{L}{2} - d)$ by $(\frac{L}{2} - d)$, and $\frac{\pi d}{L}$ by $\frac{\pi(L-d)}{L}$

$$u_3 = c_{\ell g}\ [W(\tfrac{L}{2})\ \dot{b} + \dot{y}_c + h_T\ \dot{\theta}_{co} - \dot{y}_g] + k_{\ell g}\ [W(\tfrac{L}{2})\ b + h_T\ \theta_{co} + y_c - y_g]$$

. Mass Matrix

$\underline{M}$ = same as in the previous case of 11 isolator forces

. Stiffness Matrix, $\underline{K}$

$$K_{11} = (EI)_L \beta_1{}^4 L$$

$$K_{12} = \ldots = K_{1,10} = 0$$

$$K_{21} = \ldots = K_{2,10} = 0$$

$$K_{31} = K_{32} = 0$$

$$K_{33} = \frac{k_s B^2}{2}$$

$$K_{34} = \frac{k_s B^2}{4}\ [\cos \frac{\pi d}{L} + \cos \frac{\pi (L - d)}{L}] = K_{43}$$

$$K_{35} = \ldots = K_{38} = 0$$

$$K_{39} = \frac{-k_s B^2}{4} = K_{93} = K_{3,10} = K_{10,3} = K_{49} = K_{4,10} = K_{10,4} = K_{94}$$

$$K_{44} = \frac{k_s B^2}{4}\ [\cos^2 \frac{\pi d}{L} + \cos^2 \frac{\pi (L - d)}{L}] + \frac{GJL}{2}(\frac{\pi}{L})^2$$

$$K_{45} = \ldots = K_{48} = 0$$

$$K_{51} = \ldots = K_{5,10} = 0$$

$$K_{61} = \ldots = K_{6,10} = 0$$

$$K_{71} = \ldots = K_{76} = 0$$

$$K_{77} = \ldots = k_{\ell r}$$

$$K_{78} = K_{87} = 0$$

$$K_{88} = k_{\ell f}$$

$$K_{89} = K_{98} = 0$$

$$K_{8,10} = K_{10,8} = k_{\ell f} h_{1f}$$

$$K_{91} = K_{92} = K_{95} = K_{96} = K_{9,10} = 0$$

$$K_{97} = K_{79} = k_{\ell r} h_{1r}$$

$$K_{99} = \frac{k_s \; B^2}{4} + \frac{k_{Ar} R^2}{4} + k_{\alpha r} + h_{1r}^2 K_{\ell r}$$

$$K_{10,10} = \frac{k_s B^2}{4} + \frac{K_{Af} R^2}{4} + k_{\alpha f} + k_{\ell f} h_{1f}^2$$

. Damping Matrix, C

Identical in form to $\underline{K}$ except $C_{11} = 0$,

$$C_{44} = \frac{c_s B^2}{4} \left[\cos^2 \frac{\pi d}{L} + \cos^2 \frac{\pi (L - d)}{L}\right]$$

For other elements k_s, k_A, k_α are replaced by c_s, c_A, c_α respectively.

. Control Force Matrix, U

$$U_{11} = W(d) \qquad U_{52} = \frac{L}{2} - d$$

$$U_{12} = W(L - d) \qquad U_{53} = U_{61} = U_{62} = 0$$

$$U_{13} = W(\tfrac{L}{2}) \qquad U_{63} = U_{71} = -1$$

$$U_{21} = U_{22} = U_{23} = 1 \qquad U_{72} = U_{73} = U_{81} = 0$$

$$U_{31} = U_{32} = h_3 \qquad U_{82} = -1$$

$$U_{33} = h_T \qquad U_{83} = 0$$

$$U_{41} = h_3 \cos \frac{\pi d}{L} \qquad U_{91} = h_{2r}$$

$$U_{42} = h_3 \cos \frac{\pi(L - d)}{L} \qquad U_{92} = U_{93} = U_{10,1} = U_{10,3} = 0$$

$$U_{43} = 0 \qquad U_{10,2} = h_{2f}$$

$$U_{51} = -(\frac{L}{2} - d)$$

. Forcing Function Matrix

$$\begin{array}{c} \\ (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \\ (7) \\ (8) \\ (9) \\ (10) \end{array} \begin{array}{c} \begin{array}{cccc} (1) & (2) & (3) & (4) \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ h_{1r} & 0 & \frac{R}{4} & 0 \\ 0 & h_{1f} & 0 & \frac{R}{4} \end{bmatrix} \end{array}$$

4. <u>Notation</u>

$A_{c\ell}(x)$	lateral acceleration of car
B	lateral distance between bolster springs (see Fig. 14)
b(t)	generalized coordinate for lateral car bending
C_{hi}	$(h-1)^{th}$ element of damping matrix
d	distance from end of car to centerline of trucks (see Fig. 13)

E	Young's modulus
f_q	frequency of q^{th} degree of freedom, $\frac{\omega_q}{2\pi}$, Hz
G	modulus of rigidity
h_1	vertical distance from traction-motor center of gravity to line of action of lateral equalizer spring (see Fig. 13(b))
h_2	vertical distance from traction-motor center of gravity to line of action of lateral bolster spring (see Fig. 13(b))
h_3	vertical distance between car elastic axis and line of action of lateral bolster spring (see Fig. 13(b))
h_4	vertical distance of car section center of gravity from car elastic axis, positive for center of gravity above elastic axis (see Fig. 13(b))
h_T	vertical distance between car elastic axis and line of action of lateral transformer spring (see Fig. 13(a))
I	flexural moment of inertia of car cross-section
I_A	mass moment of inertia of traction motor in roll about its center of gravity
I_{EA}	mass moment of inertia of car in roll about car elastic axis, per unit length
J	torsional constant of car cross-section
k	spring constant
K_{hi}	$(h-i)^{th}$ element of stiffness matrix
L	length of car (see Figs. 13(a))
M_{hi}	$(h-i)^{th}$ element of mass matrix
m	mass of car per unit length
M_A	traction-motor mass

M_c	car mass, mL
M_g	transformer mass
n_q	viscous damping coefficient for q^{th} degree of freedom, where $q = A$, $c\ell$, cT, etc., also see subscripts
R	lateral distance between equalizer springs (see Fig. 13(b))
S(t)	vertical displacement forcing function at rails (see Fig. 13(b))
v	car speed
v(x,t)	lateral bending deformation of car (see Fig. 13(a))
W(x)	car bending mode-shape value at x
x	length along car, measured from rear of car
Y(t)	lateral displacement forcing function at rails (see Fig. 13(b))
YL	lower limit
YU	upper limit
$y_A(t)$	generalized coordinate of lateral translation of traction motor
$y_c(t)$	generalized coordinate of lateral rigid-body translation of car (see Fig. 13(b))
$y_{\ell g}$	generalized coordinate of lateral transformer translation
$\delta(t)$	increment of vertical displacement at rails associated with cross-level (or rolling) displacement input (see Fig. 13(b))
$(\Delta\ell)_t$	center-to-center distance between axles of a truck suspension system
$\Delta y(t)$	displacement relation in lateral bolster springs for nonlinear spring behavior

$\eta(t)$	generalized coordinate of car in rigid-body yaw
$\theta_A(t)$	generalized coordinate of traction-motor rolling degree of freedom
$\theta_c(x,t)$	rigid-body roll and torsion of car
$\theta_{co}(t)$	generalized coordinate for rigid-car roll
$\theta_{c1}(t)$	generalized coordinate for 1st torsion mode of car

Subscripts

A	identifies properties associated with the traction-motor-equalizer system
c	associated with car
f	identifies forward trucks
g	associated with transformer
h,i	integers identifying matrix elements
ℓ	identifies properties of traction-motor-lateral-equalizer system
ℓc	identifies properties of lateral bolster spring-damper system
o	identifies amplitude of forcing function
r	identifies rear trucks
s	identifies properties of vertical bolster spring-damper system

Dots over quantities denote differentiation with respect to time.

D. Aircraft Ride Control System

To evaluate the candidate designs for the ride control system in a STOL airplane, a limiting performance problem was formulated. Specifically, it is desired to minimize the maximum vertical center-of-gravity (c.g.) acceleration subject to constraints on flap and elevator deflections. These two deflections are used as controls.

The airplane is assumed to behave as a rigid body. The nomenclature is:

α = wing angle of attack

θ = rigid-body pitch of airplane

v = perturbation in forward velocity

$\alpha_g(t)$ = angle of attack due to gust

$\alpha_g(t-T)$ = delayed gust angle of attack (at tail)

T = time delay between impingement of gust on wing and tail

u_1 = elevator angular deflection, a control function

u_2 = flap angular deflection, a control function

$\alpha_\circ$ = gust amplitude

ω = gust frequency

Assuming quasi-steady aerodynamics, the equations of motion for the airplace in first order PERFORM format are

$$\dot{\bar{s}} = \underline{A}\,\bar{s} + \underline{B}\,\bar{u} + \underline{D}\,\bar{f}_k \tag{32}$$

where

$$s = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix} = \begin{bmatrix} \theta \\ \dot{\theta} \\ \alpha \\ v \end{bmatrix} \tag{33}$$

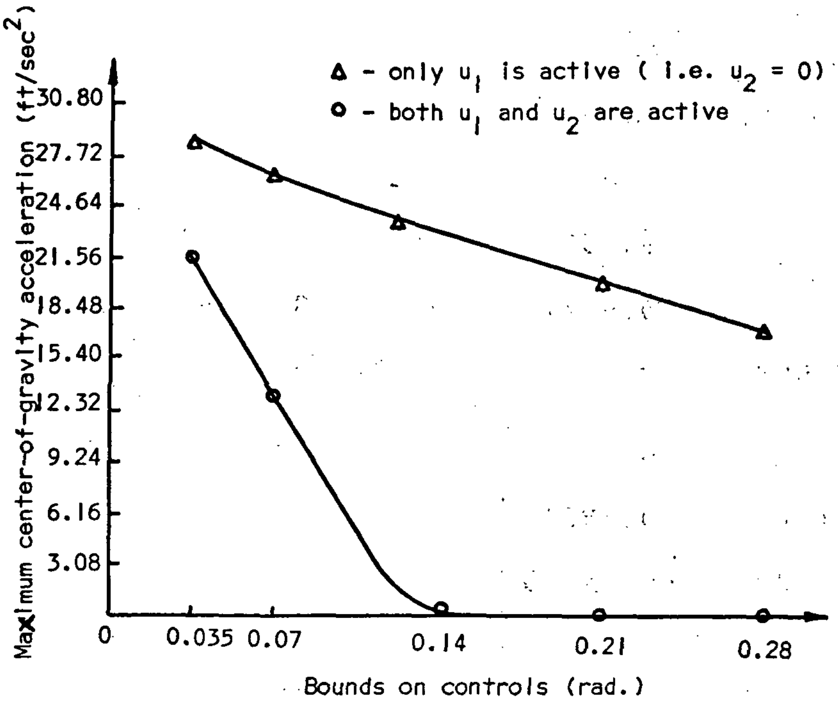


Fig. 17 Limiting Performance for a STOL Airplane Ride Control System.

$$\bar{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{34}$$

$$\bar{f}_k = \begin{bmatrix} f_{11} \\ f_{12} \end{bmatrix} = \begin{bmatrix} \alpha_g(t) \\ \alpha_g(t - T) \end{bmatrix} \tag{35}$$

Consider the specific case:

$$\underline{A} = \begin{bmatrix} 0 & 1.0 & 0 & 0 \\ 0 & -0.05662 & -0.58024 & -0.0000262 \\ 0 & 0.97710 & -1.75148 & -0.001 \\ -4.77992 & 0 & 0.96138 & 0 \end{bmatrix}$$

$$\underline{B} = \begin{bmatrix} 0 & 0 \\ -0.64961 & -0.18396 \\ -0.15183 & -0.64855 \\ 0 & 0 \end{bmatrix}$$

$$\underline{D} = \begin{bmatrix} 0 & 0 \\ -0.13587 & -0.10674 \\ -1.65057 & -0.14425 \\ 0 & 0 \end{bmatrix}$$

Select $v_o(\dot{s}_1 - \dot{s}_3)$ as the objective function, with v_o= 308 ft/sec. The constraints are imposed on the magnitude of the control. Numerical solutions are obtained for two cases. In Case 1 only u_1 is active and $u_2 = 0$. In Case 2, both u_1 and u_2 are active. The disturbance in both instances is a step input, i.e., $\alpha_g(t) = 0.649H(t)$ where $H(t)$ is a unit step function. The time delay is $T = .08207$ sec. The total time interval used was 0.95 sec with a time increment of 0.05 sec. The trade-off curves for these two cases are shown in Fig. 17 .

E. Shock Absorbers for Freight Cars

Consider the limiting performance of the shock-absorber system (or cushion) of railroad vehicles used in protecting passengers or cargos under crash conditions. Typical analyses of freight car lading protection problems are described in Refs. 4 and 5.

The dynamic system model of Fig. 18 is taken from Ref. 5, with the following nomenclature:

S_o = lading contact area

c = damping coefficient, due to friction in lading

k = elastic constant, representing resilience in lading

M_c = mass of struck car body

M_L = mass of lading

M_s = mass of striking car

u = cushion force

$d = y_1 - y_2$ = cushion travel

V = impact velocity

The equations of motion are given by

$$M_s \ddot{y}_1 + u = 0$$

$$M_c \ddot{y}_2 - k(y_3 - y_2) - c(\dot{y}_3 - \dot{y}_2) - u = 0 \qquad (36)$$

$$M_L \ddot{y}_3 + k(y_3 - y_2) - c(\dot{y}_3 - \dot{y}_2) = 0$$

We will treat the impact conditions

$$y_1(0) = y_2(0) = \dot{y}_2(0) = 0$$

$$\dot{y}_2(0) = V$$

where $t = 0$ is the time at which impact occurs.

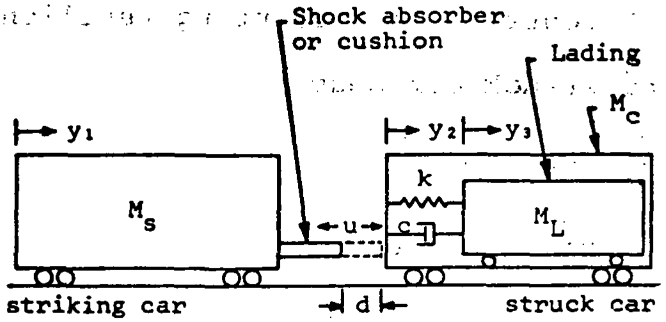


Fig 18. Model of striking car, struck car and lading.

For specified impact velocities, the problem is to minimize the peak force transmitted to the lading while the cushion travel distance is bounded. Thus, the u is sought that minimizes

$$\max|k(y_3 - y_2)| \tag{37}$$

with the restriction that

$$0 \leq d \leq A \tag{38}$$

where A is prescribed.

The problem has been put in PERFORM format for the following physical and material constraints:

$c = 23{,}000$ lb sec/ft

$k = 341{,}667$ lb/ft

$M_c = 1630$ lb sec^2/ft

$M_L = 1590$ lb sec^2/ft

$M_s = 5280$ lb sec^2/ft

$S_o = 33.7$ ft^2

$V = 10$ mph

A PERFORM generated trade-off diagram between maximum transmitted force and cushion travel distance is shown in Fig. 19.

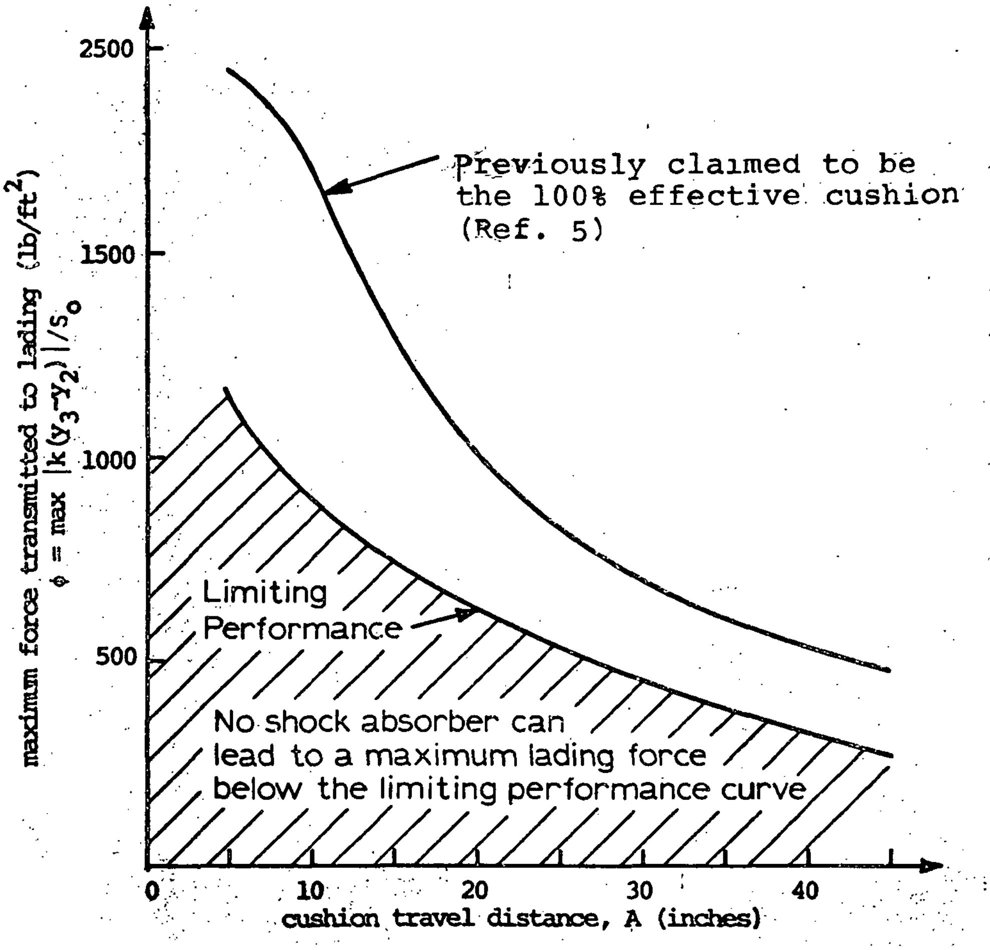


Fig 19. Trade-off diagram for rail vehicle cushioning system

F. A Launch Booster Control Problem

We consider here a booster control problem known as the constrained bending moment minimax drift problem. This is the problem of minimizing the lateral drift from a reference trajectory along a flight path where the magnitude of the bending moment cannot exceed a critical value (Refs. 6, 7). This problem will be placed in the PERFORM format in the following subsections.

1. Equations of Motion

The rigid body equations of a typical launch booster are given in the following form (Ref. 6:

$$\ddot{\phi} + C_1\alpha + C_2\beta = 0 \tag{39}$$

$$\ddot{Z} = K_1\phi + K_2\alpha + K_3\beta \tag{40}$$

$$\alpha = \phi - \frac{\dot{Z}}{V} + \alpha_\omega \tag{41}$$

where ϕ, α, β, Z represent angular deviation from reference, angle of attack, engine gimble angle, and lateral positional deviation from reference, respectively. The wind contribution to the angle of attack is denoted by α_ω and the vehicle speed by V. We assume several transient wind disturbance profiles which might be encountered by the launch vehicle during flight are known.

By substituting Eq. (41) into Eqs. (39) and (40), we have

$$\ddot{\phi} = -C_1\phi + \frac{C_1}{V}\dot{Z} - C_1\alpha_\omega - C_2\beta$$

$$\ddot{Z} = (K_1 + K_2)\phi - \frac{K_2}{V}\dot{Z} + K_2\alpha_\omega + K_3\beta$$

Let

$$\bar{s} = \begin{bmatrix} \dot{\phi} \\ \dot{Z} \\ \phi \\ Z \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

and $u = \beta$, and $f_k = \alpha_\omega$

Then the rigid body equations of motion for the booster take the PERFORM compatible form

$$\dot{\bar{s}} = \underline{A}\,\bar{s} + \underline{B}\,\bar{u} + \underline{D}\,\bar{f}_k$$

where

$$\underline{A} = \begin{bmatrix} 0 & \frac{C_1}{V} & -C_1 & 0 \\ 0 & \frac{-K_2}{V} & K_1+K_2 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\underline{B} = \begin{bmatrix} -C_2 \\ K_3 \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \underline{D} = \begin{bmatrix} -C_1 \\ K_2 \\ 0 \\ 0 \end{bmatrix}$$

2. Objective Function and Constraints

The bending moment, given by a linear combination of α and β is constrained as follows:

$$\left| M'_\alpha\,\alpha + M'_\beta\,\beta \right| \leq M_{max} \qquad (42)$$

where M'_α and M'_β are bending moment coefficients and M_{max} is the maximum allowable bending moment. We choose as an objective the minimization of the maximum $|\dot{z}\ (t)|$ of the booster subject to the constraint given by Eq. (42).

In the PERFORM format the objective function is written as

$$\underline{PX1}\bar{s} + \underline{PX2}\bar{u} + \underline{PX3}\ \bar{f}_k$$

where

$$\underline{PX1} = [\ 0\ 1\ 0\ 0\ ]$$

$$\underline{PX2} = [\ 0\ ]$$

$$\underline{PX3} = [\ 0\ ]$$

In the PERFORM input format the constraint given by Eq. (42) can be expressed as

$$\overline{YL} \leq \underline{Y1}\bar{s} + \underline{Y2}\bar{u} + \underline{Y3}\ \bar{\bar{f}}_k \leq \overline{YU}$$

where

$$\overline{YL} = -|M_{max}|$$

$$\overline{YU} = |M_{max}|$$

$$\underline{Y1} = [0\ \ -\frac{M'_\alpha}{V}\ \ M'_\alpha\ \ 0]$$

$$\underline{Y2} = [M'_\beta]$$

$$\underline{Y3} = [M'_\alpha]$$

The specification matrix is given by

$$MSP = [\ 0\ \ 0\ ]$$

3. Output

The rate of change of the lateral positional deviation from reference, $\dot{z}(t)$, can be tabulated versus time. Also, a trade-off relation between the minmax $\dot{Z}(t)$ and the maximum bending moment can be obtained by varying the bounds of the bending moment.

G. A Pressurized-Water Nuclear Reactor Power Plant

As a potential application of the computer program to a large system, we consider the control of a pressurized-water nuclear power plant. This system was first used as an example in Ref. [8]. Fig. 20 shows the schematic diagram of the pressurized water reactor. Heat is produced in the reactor by nuclear fission, with the rate controlled by insertion and withdrawal of control rods. This heat is carried away by the primary coolant which then flows to a heat exchanger where it gives up its heat to make steam. The steam goes to a turbine that converts the energy to shaft work to produce electric power. Pumps cause the primary coolant to circulate through the primary loop.

A pressurizer rides above the primary loop with saturated water in the lower part and saturated steam in the upper. Having saturated conditions in the pressurizer assures that the water in the primary loop is subcooled. The steam in the upper part of the pressurizer acts as a "cushion" to allow expansion or contraction of the water as its temperature varies. Spray water taken from the coldest part of the primary loop condenses steam and lowers the pressure. An electric heater boils the saturated water, thus lowering the water level and increasing pressure.

The principal disturbance to the plant operation comes from sudden changes in steam demand from electrical load changes. The most difficult case is a load drop from maximum load. The reduced steam flow takes less heat from the heat exchanger, causing temperature and pressure to rise rapidly in the primary loop.

A ten percent step drop in steam flow from full load is applied with the plant initially at equilibrium. This load change is treated as a forcing function.

The objective is to maximize the minimum value of the pressure of the pressurizer with continuous constraints on maximum pressure and the three controls plus terminal constraints on a temperature and its rate of change. We sketch the formulation of this problem in the PERFORM format in the following subsections. The specific coefficients involved can be found in Ref. 8.

1. Governing Equations

The equations describing the system are expressed as

$$\dot{\bar{s}} = \underline{A}\,\bar{s} + \underline{B}\,\bar{u} + \underline{D}\,\bar{f}_k$$

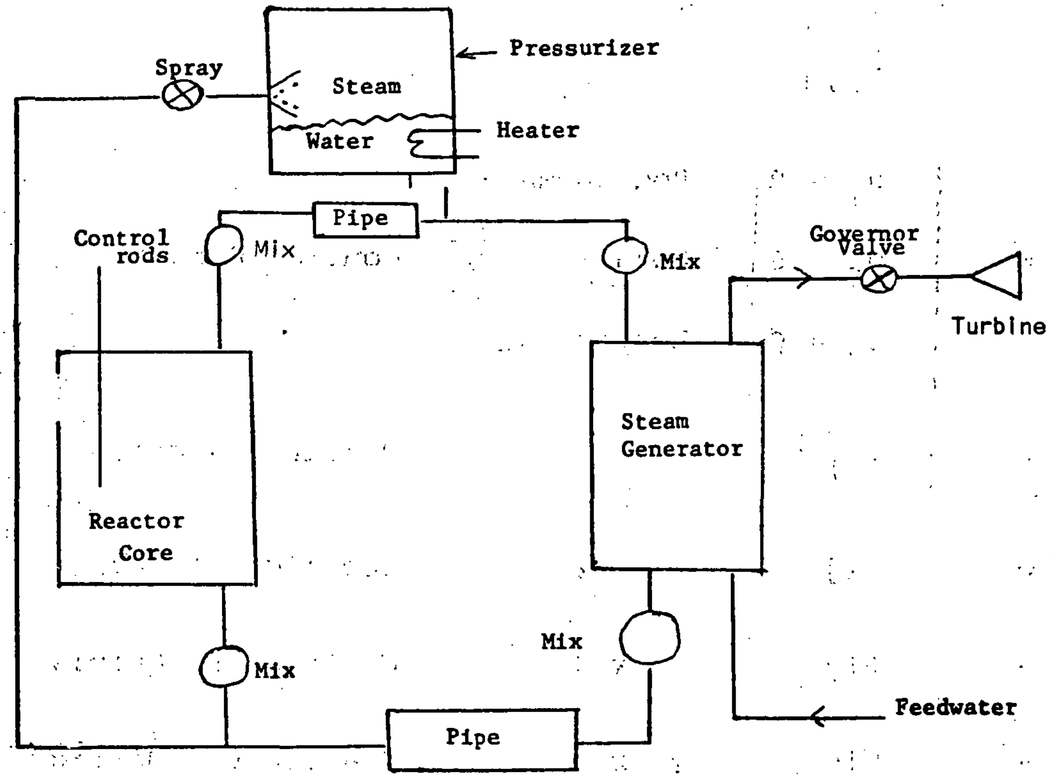


Fig. 20 Schematic Diagram of Pressurized-Water Reactor

where

$$\bar{s} = \begin{bmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ s_{25} \end{bmatrix} \quad \text{state vector}$$

$$\bar{u} = \begin{bmatrix} u_1 = R \\ u_2 = S \\ u_3 = Q \end{bmatrix} \quad \left.\begin{array}{l} \text{Rod speed} \\ \text{Spray} \\ \text{Heater} \end{array}\right\} \text{Control Forces}$$

$$\bar{f}_k = \begin{bmatrix} f_{11} \end{bmatrix} \qquad \text{Forcing Function}$$

$$\underline{A} = \begin{bmatrix} Aij \end{bmatrix} \quad 25 \times 25 \qquad \text{Coefficient Matrix}$$

$$\underline{B} = \begin{bmatrix} Bij \end{bmatrix} \quad 25 \times 3 \qquad \text{Coefficient Matrix}$$

$$\underline{D} = \begin{bmatrix} Dij \end{bmatrix} \quad 25 \times 1 \qquad \text{Coefficient Matrix}$$

2. Objective Function

The objective function is expressed in the matrix form

$$\underline{PX1}\ \bar{s} + \underline{PX2}\ \bar{u} + \underline{PX3}\ \bar{f}_k$$

where

$\bar{s}$, $\bar{u}$, and $\bar{f}_k$ are defined above.

$$\underline{PX1} = \begin{bmatrix} PX_{ij} \end{bmatrix} \quad 1 \times 25 \qquad \text{Coefficient Matrix}$$

$$\underline{PX2} = \begin{bmatrix} PX_{ij} \end{bmatrix} \quad 1 \times 3 \qquad \text{Coefficient Matrix}$$

$\underline{PX3} = \left[PX_{ij} \right]$ 1 x 1 Coefficient Matrix

3. Constraints

Constraints in matrix form are expressed as

$$\overline{YL} \leq \underline{Y1}\bar{s} + \underline{Y2}\bar{u} + \underline{Y3}\bar{f}_k \leq \overline{YU}$$

where

$\bar{s}$, $\bar{u}$, and $\bar{f}_k$ are defined above.

$\overline{YL} =$	$\left[YL_{ij} \right]$	6 x 1	Lower Limit
$\underline{Y1} =$	$\left[Y1_{ij} \right]$	6 x 25	Coefficient Matrix
$\underline{Y2} =$	$\left[Y2_{ij} \right]$	6 x 3	Coefficient Matrix
$\underline{Y3} =$	$\left[Y3_{ij} \right]$	6 x 1	Coefficient Matrix
$\overline{YU} =$	$\left[YU_{ij} \right]$	6 x 1	Upper limit

The specification matrix is given by

$$MSP = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 3 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

4. Output

The desired output vector $\bar{V}$ is given by

$$\bar{V} = \underline{Q1}\ \bar{s} + \underline{Q2}\ \bar{u} + \underline{Q3}\ \bar{f}_k$$

where

$$\bar{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \end{bmatrix} \begin{matrix} \text{Pressure} \\ \text{Temperature} \\ \text{Time derivative of temperature} \\ \text{Rod speed} \\ \text{Spray} \\ \text{Heater} \end{matrix}$$

$\underline{Q1} = \left[Q_{ij} \right]$ 6 x 25 Coefficient Matrix

$\underline{Q2} = \left[Q_{ij} \right]$ 6 x 3 Coefficient Matrix

$\underline{Q3} = \left[Q_{ij} \right]$ 6 x 1 Coefficient Matrix

Trade-off relations between the minimum value of the pressurizer pressure and the output vector $\underline{V}$ shown above can be obtained by varying the bounds $\overline{YL}$ and $\overline{YU}$.

SECTION V

CONCLUSIONS

A computational capability, PERFORM, for the evaluation of the limiting performance of transient dynamic systems has been developed and is described in this report. This capability is intended to be used by a designer of dynamic systems to determine the feasibility of the proposed design specifications. Thus, PERFORM is meant to be a new design tool.

In addition to being able to use PERFORM to scrutinize design specifications, this capability can be used during the actual design process to measure the relative success of proposed designs with the characteristics of the theoretically best design provided by PERFORM.

PERFORM is used in an engineering problem-oriented form. The user prescribes in a simple fashion the system equations of motion, sets of constraints, objective functions, and classes of possible inputs. The capability then automatically computes the limiting performance characteristics requested by the user.

SECTION VI

REFERENCES

1. Sevin, E., and Pilkey, W.D. Optimum Shock and Vibration Isolation, Monograph No. 6., Shock and Vibration Information Center, Washington, D.C., 1971.

2. Klein, G.H. "Optimization of Linear, Nonlinear and Active Multi-Degree-of-Freedom Shock Isolation Systems," Ph.D. Dissertation, University of California, Los Angeles, 1971.

3. Sewall, J.L., Parrish, R.V., and Durling, B.J. "Dynamic Responses of Railroad Car Models to Vertical and Lateral Rail Inputs," NASA TN D-6375, 1971.

4. Freudenstein, F. "Dynamic Analysis of Long-Travel, High-Efficiency Shock Absorbers in Freight Cars," ASME Paper No. 70-RR-1, Journal of Engineering for Industry, 1970.

5. Peterson, W.H. "Cushioning Requirements for Adequate Lading Protection," ASME Paper No. 59-A-312, 1959.

6. Graham, K.D. "Minimax Control of Large Launch Boosters," NASA CR-546, 1966.

7. Davis, K.J. and Ford, D.A. "Studies in Minimax Control," NASA TM-X-53189, 1964.

8. Enns, M.K. "Optimal Control of Multivariable Dynamic Systems Through Linear Programming," Ph.D. Thesis, University of Pittsburgh, 1967.

9. Hastings, C., Jr. Approximations For Digital Computors, Princeton University Press, Princeton, N.J., 1955.

APPENDIX I

USERS GUIDE FOR PERFORM

PERFORM is a computer system that can be used to determine the limiting performance characteristics of a dynamic system described by a system of first or second order equations of motion. The information, including input requirements, necessary to use PERFORM is described in this appendix.

A. PERFORM Utilization Procedure

This section describes the steps necessary to solve a limiting performance problem with the PERFORM system. Essentially this is a statement of the details of the system flowchart of Fig. 1, Section III.

1. Prepare the limiting performance problem specification data. This input is described in detail in the following sections of this appendix. In the system flowchart this input is labeled "PERFORM problem specification data."

2. Run PREPROC using the data of Step 1. There are two card outputs of this program; the PSTPROC Report data and the linear programming problem data for the LP Solver.

   Hold the PSTPROC report data output for use in Step 6.

   If several sets of constraints are being run for trade-off relations, one LP Solver data deck will be produced for each different constraint.

   Deck preparation for compiling and executing the PREPROC program is shown in Fig. I-1. The details of the leading control cards depend on the operating system of the computer and any user desired system option.*

3. Transmit the linear programming program data to the external computer, i.e. the computer with either MPS/360 or a MPS/360 compatible linear programming software, for solution of the linear programming problem.

---

*If recurring use of PERFORM were anticipated, excessive recompilation can be avoided by storing the program, in object form, on the system library. The indicated deck structure would then have to be appropriately modified.

4. Run MPS/360, or compatible software, to obtain the problem solution. For each constraint deck produced one execution of MPS is required; however, only one job and transmission is required.

   The deck contents for an MPS/360 run is shown in Fig. I-2. This model, in general, would be correct for any MPS/360 installation using OS/360. Words in upper case letters are invariant among jobs and installations. Words in lower case letters depend on the job and installation. Those dependent on the job are user chosen to identify the problem. Those dependent on the installation should be obtained from system personnel.

   Because of possible idiosyncrasies of the external computer, consultation with system personnel before attempted use is imperative.

5. Receive from the external computer the linear programming problem solution. This can be either in card form via courier or by telecommunications if facilities are available.

6. Run *PSTPROC* using the *MPS/360 linear* programming solution data received in Step 5 and the PSTPROC report data of Step 1.

   Fig. I-3 shows the deck structure for this run. The details of the leading control cards depend on the operating system being used and any user desired system options.*

   The different linear programming solution decks must be physically concentrated behind the PSTPROC data. The importance of careful card handling cannot be understated since the existing PERFORM system has no facilities for error detection.

   The limiting performance problem solution is output in this step.

---

*As indicated in the discussion of Step 2, it may be advisable to place the PSTPROC program on the system library.

Job and other Control Cards

↑

PREPROC program

↓

Data Indicator (CDC 6000: 7-8-9 card)

Problem Specification Data

↓

End of Job Indicator (CDC 6000: 6-7-8-9 card)

(Lines Represent Decks of Cards)

Figure I-1 PREPROC Deck Structure.
Compile and Execute

```
//  Job-name           JOB         account data

//                     EX          MPS-procedure-name

//  Compile-step-name. SYSIN             DD         *

              MPS/360   CONTROL     PROGRAM

/*

//  Go-stepname.SYSIN                    DD         *

              MPS/360 LINEAR PROGRAMMING DATA

/*
```

(Lines Represent Decks of Card)

Figure I-2 MPS/360 Execution Deck

Job and other Control Cards

PSTPROC program

Data Indicator (CDC 6000: 7-8-9 card)

PSTPROC Report Deck

MPS/360 SOLUTION Deck

(REPEATED)

End of job indicator (CDC 6000: 6-7-8-9 card)

(Lines Represent Decks of Card)

Figure I-3 PSTPROC Deck Structure

Compile and Execute

B. Input for PERFORM

1. System Described by Second Order Equations

PERFORM treats svstems described by the second order equations of motion

$$\sum_{j=1}^{n} M_{ij}\ddot{q}_j + \sum_{j=1}^{n} C_{ij}\dot{q}_j + \sum_{j=1}^{n} K_{ij}q_j + \sum_{j=1}^{nu} U_{ij}u_j = \sum_{j=1}^{nf} F_{ij}f_{kj} \quad \text{(I-1)}$$

$i = 1$ to n

in matrix form these become

$$\underline{M}\,\ddot{\bar{q}} + \underline{C}\,\dot{\bar{q}} + K\,\bar{q} + \underline{U}\,\bar{u} = \underline{F}\,\bar{f}_k \quad \text{(I-2)}$$

where

$$\bar{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad n \times 1 \qquad \text{displacement vector}$$

$$\dot{\bar{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad n \times 1 \qquad \text{velocity vector}$$

$$\ddot{\bar{q}} = \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \\ \vdots \\ \ddot{q}_n \end{bmatrix} \quad n \times 1$$

acceleration vector

$$\bar{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{nu} \end{bmatrix} \quad nu \times 1$$

control force vector

$$f_k = \begin{bmatrix} f_{k1} \\ f_{k2} \\ \vdots \\ f_{knf} \end{bmatrix} \quad nf \times 1$$

forcing function vector. The subscript k identifies the set of forcing functions, e.g. f_{21} is the first element of the second set of forcing functions. This allows the system to encounter alternate sets of forcing functions which might occur with equal probability.

$$\underline{M} = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & M_{2n} \\ \vdots & & & \\ M_{n1} & M_{n2} & \cdots & M_{nn} \end{bmatrix} = [M_{ij}]_{n \times n}$$

mass matrix

$\underline{C} = [C_{ij}]_{n \times n}$ — damping matrix

$\underline{K} = [K_{ij}]_{n \times n}$ — spring matrix

$\underline{U} = [U_{ij}]_{n \times nu}$ — matrix of constants associated with control force

$\underline{F} = [F_{ij}]_{n \times nf}$ — matrix of constants associated with external disturbances

The user should place his equations of motion in the form of Eq. (I-2). The non-zero elements of the matrices $\underline{M}$, $\underline{C}$, $\underline{K}$, $\underline{U}$, $\underline{F}$ are then entered as inputs in the form explained subsequently. This is accomplished by identifying the matrix, e.g., M MATRIX, and then specifying an element and its value, e.g., i, j and M_{ij}. Elements not entered are assumed to be zero.

Example 1

The equations of motion for a system shown in Fig. I-4 can be written as

$$m_1\ddot{x} + k(x-y) + u = m_1 g + h_k(t) \tag{I-3}$$

$$m_2\ddot{y} + c\dot{y} + k(y-x) = m_2 g$$

Put in matrix form

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & c \end{bmatrix}\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}[u] = \begin{bmatrix} m_1 & 1 \\ m_2 & 0 \end{bmatrix}\begin{bmatrix} g \\ h_k(t) \end{bmatrix} \tag{I-4}$$

Now define

$$x = q_1 \qquad y = q_2$$

$$\bar{q} = \begin{bmatrix} x \\ y \end{bmatrix}$$

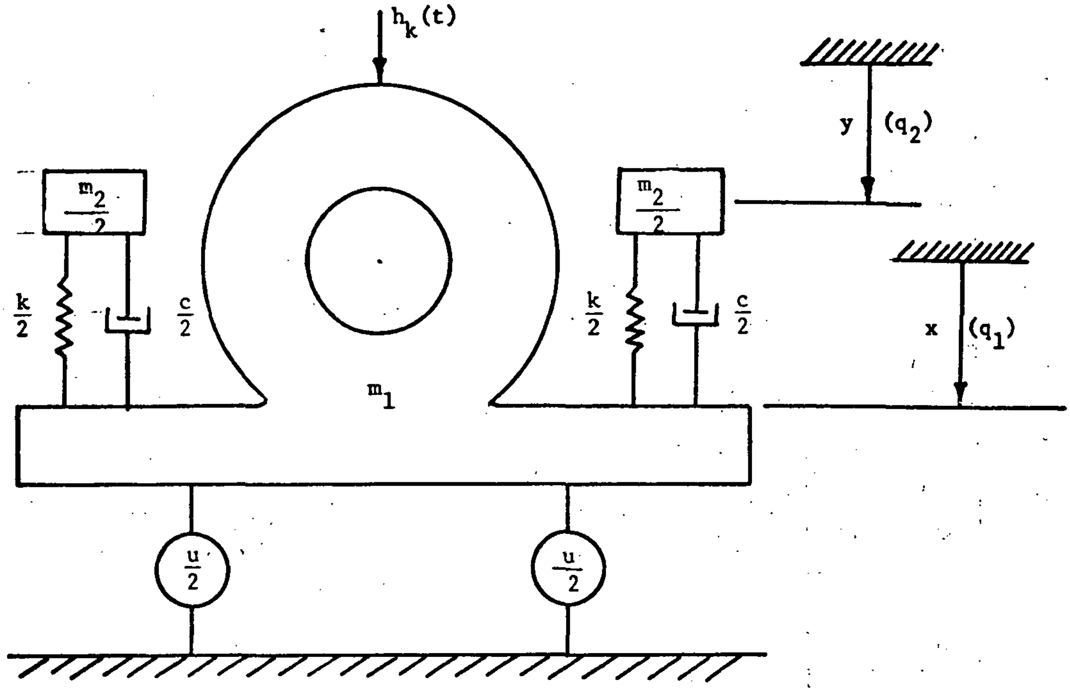

$h_k(t)$
$\frac{m_2}{2}$
$\frac{m_2}{2}$
$\frac{k}{2}$
$\frac{c}{2}$
$\frac{k}{2}$
$\frac{c}{2}$
m_1
y
(q_2)
x
(q_1)
$\frac{u}{2}$
$\frac{u}{2}$

Fig. I-4

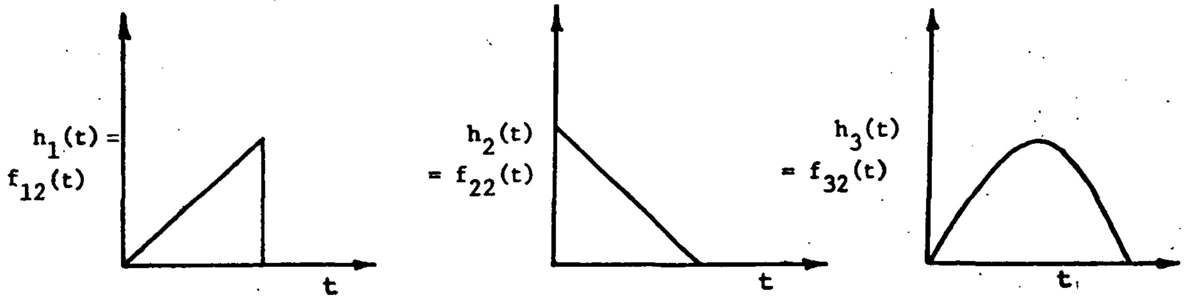

$h_1(t) =$
$f_{12}(t)$
t
$h_2(t)$
$= f_{22}(t)$
t
$h_3(t)$
$= f_{32}(t)$
t

Fig. I-5

$$\underline{\dot{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

$$\underline{\ddot{q}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}$$

$$\bar{u} = \begin{bmatrix} u \end{bmatrix}$$

$$\underline{M} = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \qquad \text{(I-5)}$$

$$\underline{C} = \begin{bmatrix} 0 & 0 \\ 0 & c \end{bmatrix}$$

$$\underline{K} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix}$$

$$\underline{U} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$\underline{F} = \begin{bmatrix} m_1 & 1 \\ m_2 & 0 \end{bmatrix}$$

$$\bar{f}_k = \begin{bmatrix} g \\ h_k(t) \end{bmatrix}$$

Note that the k subscript of $\bar{f}$ designates one set of forcing functions. In our case $f_{k1} = g$ = gravitational acceleration constant while $f_{k2} = h_k(t)$ = some function of time, e.g. those in Fig. I-5. If f_{12}, f_{22}, and f_{32} of Fig. I-5 were all to be included in the input, then these disturbances would be treated as though all are potential forcing functions that may occur with equal probability. Using Eqs. (I-5), Eq. (I-4) becomes

$$\underline{M}\,\ddot{\bar{q}} + \underline{C}\,\dot{\bar{q}} + \underline{K}\,\bar{q} + \underline{U}\,\bar{u} = \underline{F}\,\bar{f}_k \tag{I-6}$$

2. <u>System Described by First Order Equations</u>

The first order equations of motion accepted by the computer system are expressed as

$$\dot{\bar{s}} = \underline{A}\bar{s} + \underline{B}\bar{u} + \underline{D}\bar{f}_k \tag{I-7}$$

where

$$\bar{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{bmatrix} \quad m \times 1 \qquad \text{state vector}$$

$$\dot{\bar{s}} \quad \begin{bmatrix} \dot{s}_1 \\ \dot{s}_2 \\ \vdots \\ \dot{s}_m \end{bmatrix} \quad m \times 1 \qquad \text{time derivative of state vector}$$

$\underline{A} = [A_{ij}]_{m \times m}$ coefficient matrix

$\underline{B} = [B_{ij}]_{m \times nu}$ coefficient matrix

$\underline{D} = [D_{ij}]_{m \times nf}$ coefficient matrix

m = number of equations in the system of equations

The user should input non-zero elements of matrices $\underline{A}$, $\underline{B}$, $\underline{D}$ in the form explained later. This is accomplished by identifying the matrices; e.g., A MATRIX, and then specifying an element and its value, e.g., i, j and A(i,j). Elements not entered are assumed to be zero.

Example 2

Equations of motion for the system shown in Fig. I-6 can be written as

$$C_1 \frac{dh_1}{dt} = -\frac{h_1 - h_2}{R_1} + e(t)$$

$$C_2 \frac{dh_2}{dt} = \frac{h_1 - h_2}{R_1} - \frac{h_2}{R_2} + E_k(t) \quad \text{or}$$

$$\frac{dh_1}{dt} = \frac{-1}{R_1 C_1} h_1 + \frac{1}{R_1 C_1} h_2 + \frac{1}{C_1} e(t) \tag{I-8}$$

$$\frac{dh_2}{dt} = \frac{1}{R_1 C_2} h_1 - \left(\frac{1}{R_1 C_2} + \frac{1}{R_2 C_2}\right) h_2 + \frac{1}{C_2} E_k(t)$$

The system of Eqs. (I-8) is equivalent to

$$\dot{\bar{s}} = \underline{A}\bar{s} + \underline{B}\bar{u} + \underline{D}\bar{f}_k \tag{I-9}$$

where

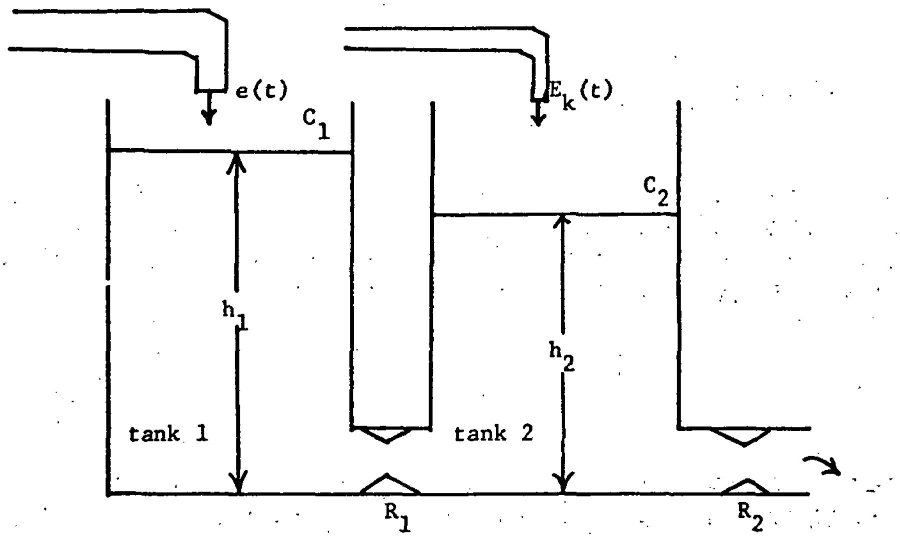
$$\bar{s} = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$$

$$\bar{u} = [e(t)]$$

$$\bar{f}_k = [E_k(t)]$$

$$\underline{A} = \begin{bmatrix} -\frac{1}{R_1 C_1} & \frac{1}{R_1 C_1} \\ \frac{1}{R_1 C_1} & -\left(\frac{1}{R_1 C_2} + \frac{1}{R_2 C_2}\right) \end{bmatrix}$$

$$\underline{B} = \begin{bmatrix} \frac{1}{C_1} \\ 0 \end{bmatrix}$$



h_1, h_2 = liquid level in tank 1,2 respectively

C_1, C_2 = cross sectional area of tank 1, 2 respectively

R_1, R_2 = orifice resistances

Flow rate through orifice = $\frac{h}{R}$ (assumption)

Fig. I-6 Two-tank liquid-level system

$$\underline{D} = \begin{bmatrix} 0 \\ \frac{1}{C_2} \end{bmatrix}$$

3. Objective Function

Regardless of the form (first or second order) used to describe the equations of motion, the format for the objective function is the same. In the case of the second order equations, the user should establish a state variable vector $\bar{s}$ as

$$\bar{s} = \begin{bmatrix} \dot{\bar{q}} \\ \cdots \\ \bar{q} \end{bmatrix} \qquad \text{(I-10)}$$

The user can choose any linear combination of state variables, derivatives of state variables, or control forces as an objective function. In the case of the system described by second order equations, these become linear combinations of accelerations, velocities, displacements and control forces. PERFORM finds the vector $\bar{u}$ that minimizes the maximum (ISP = 0) (or maximizes the minimum (ISP = 1)) time values of the functions used as objective functions while satisfying the constraints. The objective function is formed as

$$\underline{PX1}\bar{s} + \underline{PX2}\bar{u} + \underline{PX3}\bar{f}_k \qquad \text{(I-11)}$$

If more than one row of the matrices in Eq. (I-11) contains non-zero elements, then the peak values in time of the vectors resulting from the meaningful rows are to be compared. PERFORM minimizes (maximizes) the maximum (minimum) of the peak values.

Example 3

To illustrate the formation of an objective function, consider the 2-degree-of-freedom (DOF) system in Example 1. The equations of motion are given as Eqs. (I-3). Suppose we have only one set of forcing functions. We assume the objective function is to minimize the maximum of the peak values of $|\ddot{x}|$; $|\ddot{y}|$; $|\dot{x} - \dot{y}|$. From Eqs. (I-3) $\ddot{y}$ and $\ddot{x}$ can be expressed as

$$\ddot{x} = \frac{-k}{m_1} x + \frac{k}{m_1} y - u + g + \frac{h_k(t)}{m_1}$$

$$\ddot{y} = \frac{-c}{m_2} \dot{y} + \frac{k}{m_2} x - \frac{k}{m_2} y + g$$

Then the objective function involves

$$\left| -\frac{k}{m_1} x + \frac{k}{m_1} y - u + g + \frac{h_k(t)}{m_1} \right|$$

$$\left| -\frac{c}{m_2} \dot{y} + \frac{k}{m_2} x - \frac{k}{m_2} y + g \right| \qquad \text{(I-12)}$$

$$\left| \dot{x} - \dot{y} \right|$$

PERFORM will minimize the maximum of peak values of these three relations. In terms of the form required for PERFORM, we write

$$\bar{s} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ x \\ y \end{bmatrix}$$

$$\bar{u} = [u]$$

$$\bar{f}_k = \begin{bmatrix} g \\ h_k(t) \end{bmatrix}$$

so that the objective function becomes

$$\begin{bmatrix} 0 & 0 & \frac{-k}{m_1} & \frac{k}{m_1} \\ 0 & -\frac{c}{m_2} & \frac{k}{m_2} & \frac{-k}{m_2} \\ 1 & -1 & 0 & 0 \end{bmatrix} \bar{s} + \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \bar{u} + \begin{bmatrix} 1 & \frac{1}{m_1} \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \bar{f}_k \qquad \text{(I-13)}$$

Thus in Eq. (I-11),

$$\underline{PX1} = \begin{bmatrix} 0 & 0 & \frac{-k}{m_1} & \frac{k}{m_1} \\ 0 & \frac{-c}{m_2} & \frac{k}{m_2} & \frac{-k}{m_2} \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

$$\underline{PX2} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}$$

$$\underline{PX3} = \begin{bmatrix} 1 & \frac{1}{m_1} \\ 1 & 0 \\ 0 & 0 \end{bmatrix}$$

4. Constraints

Constraints, like objective functions, are treated the same regardless of whether the equations of motion are expressed in first or second order form. Constraints may be placed on state variables, derivatives of state variables, and control forces (accelerations, velocities, displacements and control forces for systems described by second order equations). The general form is

$$\overline{YL} \leq \underline{Y1}\bar{s} + \underline{Y2}\bar{u} + \underline{Y3}\bar{f}_k \leq \overline{YU} \qquad \text{(I-14)}$$

Each row in Eq. (I-14) represents one constraint. A specification matrix (MSP) is needed to specify the nature of each constraint. MSP is a matrix of two columns with rows equal in numbers to the rows in Eq. (I-14). The elements of MSP are given integer values defined as follows:

$$
MSP(I,1) = \begin{cases} 0 & \text{if the } i^{th} \text{ constraint (row) is continuous (i.e., is to be imposed at many times)} \\ 1 & \text{if the } i^{th} \text{ constraint is imposed only at some specific time.} \end{cases}
$$

$$
MSP(I,2) = \begin{cases} 0 \\ 1 \\ 2 \\ 3 \end{cases} \text{means the } i^{th} \text{ constraints are of the type respectively } \begin{Bmatrix} \leq \; \leq \\ \leq \\ \geq \\ = \end{Bmatrix}
$$

<u>Example 4</u>

For the same system of Example 1 suppose we want the constraints

$$
\begin{array}{rl}
b(t) \leq u(t) \leq a(t) & \text{for all } t \\
d_1 \leq x \leq d_2 & \text{for } t_1 \text{ only} \\
y \leq d_3 & \text{for all } t \\
\dot{x}(t_2) = V_\circ & \text{for } t_2 \text{ only}
\end{array}
$$

In matrix form

$$
\begin{bmatrix} b(t) \\ d_1 \\ - \\ - \end{bmatrix} \leq \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ x \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} [u] + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} g \\ h(t) \end{bmatrix} \underset{=}{\overset{\leq}{\text{or}}} \begin{bmatrix} a(t) \\ d_2 \\ d_3 \\ V_\circ \end{bmatrix}
$$

The specification matrix is given by

$$\underline{MSP} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 3 \end{bmatrix}$$

5. Output

Any linear combination of $\bar{s}$, $\bar{u}$, $\bar{f}$ can be tabulated versus time. The user inputs matrices $\underline{Q1}$, $\underline{Q2}$, $\underline{Q3}$ in the equation

$$\underline{Q1}\ \bar{s} + \underline{Q2}\ \bar{u} + \underline{Q3}\ \bar{f}_k$$

The dimensions of matrices $\underline{Q1}$, $\underline{Q2}$, $\underline{Q3}$ are m x not, nu x not, nf x not, respectively, where m is the number of state variables and not is the number of outputs required. A specific set of forcing functions, i.e., ISET(I), must be designated for each quantity to be tabulated. The user can also get trajectories of any one of the quantities tabulated.

A trade-off curve between the objective function and any one constraint can be obtained by varying the bound of that constraint. The bounds that can be varied must be of the constant type (i.e., NCN = 0). For a constraint with both upper and lower bounds the user can either vary both bounds at some increment or keep one bound constant and only vary the other one. The details of the implementation of the constraint variation are given later in this Appendix.

Example 5

Suppose for the system in Example 2 we want to have the water level $h_1(t)$, $h_2(t)$ and the controlled input e(t) to be tabulated as functions of time. Then

$$\underline{Q1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\underline{Q2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{Q3} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

If we set ITR(1) = 1,ITR(2) = 1, ITR(3) = 1 then the time trajectories of $h_1(t)$, $h_2(t)$ and $e(t)$ will be tabulated and plotted.

Let the goal in Example 2 be to minimize the maximum value of objective functions $|\dot{h}_1(t)|$ and $|\dot{h}_2(t)|$ with the constraint $|h_1(t)| \leq D$. Then for each prescribed value of D we can obtain a minmax value of the objective functions. By varying D we can compute a tradeoff curve.

6. Input Format

Words not underlined are variables used in the code. The underlined words marked with * are required in the input. Those underlined words marked with ** are required only if the data they designate are to be entered as part of the problem. For example, for a problem with no initial conditions (i.e. all initial conditions are zero), the card with INITIAL CONDITIONS is not needed.

The order of the word-cards is immaterial as long as the appropriate data follows the proper word. For example, the K MATRIX card and its data can be placed before or after the M MATRIX card and its data.

7. Limitations as to Size of Problems

The size of problems that can be solved by the program listed in this report is restricted by the DIMENSION statements in the program. The following are the current maximum sizes of important parameters.

NU=5, NF=5, NSETS=1, II=20, NOB=3, NOC=5, NOT=10, and the combination of the above parameters should be such that

$R_H \leq 120$, $C_H \leq 40$ (See Section D of Appendix III)*.

Card No.	Columns	Variable	Format	Comments
1	1	blank		
	2-6	NPB	(I5)	problem number
	7-10	blank		
	11-70	TITLE	(12A5)	any written title or description of the problem
	71-80	blank		
2	1-5	blank		
	6	NOE	(I1)	= 1 input system of 1^{st} order equations = 2 input system of 2^{nd} order equations
	7-10	blank		
	11-15	blank		
	16-18	blank		
	19-20	NDF	(I2)	number of equations of motion
	21-23	blank		
	24-25	NU	(I2)	number of controllers of the system
	26-30	NF	(I5)	number of elements in each set of forcing functions of the system
	31-35	NSETS	(I5)	number of sets of forcing functions (= 1 even if there are no forcing functions)

* A variable dimensioned version of PREPROC is available for use on the NASA Langley Computer Operating System (CDC 6600). *Dimensional constraints are removed in an automated version of* PERFORM at the University of Virginia Computer Science Center (CDC 6400)

Card No.	Columns	Variable	Format	Comments
	36-40	II	(I5)	= 1 + number of time intervals for discretizing control forces
	41-50	TM	(F10.6)	value of time interval
	51-55	NOB	(I5)	total number of objective functions
	56-60	NOC	(I5)	total number of constraints
	61-64	blank		
	65	ISP	(I1)	= 0 minimize the maximum of objective functions = 1 maximize the minimum of objective functions
	66-70	NOT	(I5)	number of outputs
	71-80	blank		
3	1-60	ISET (1) ISET (2) ⋮ ISET (NOT)	(6I10)	where ISET (I) designates the set forcing functions required by ith output; read in up to 6 values per with additional cards as needed
	61-80	blank		

FORCING FUNCTION

Card No.	Columns	Variable	Format	Comments
4	1-16	FORCING FUNCTION**		
	17-80	blank		
4-1	1	KEY2	(I1)	= 0 continue to read data, e.g., cards 4-2 are still to be read in = 1 last data card, e,g., if NSETS = 1, then last set has been reached
	2-6	I	(I5)	the ith set of forcing functions
	7-80	blank		

Card No.	Columns	Variable	Format	Comments
4-1-1	1	KEY1	(I1)	= 0 continue to read data, e.g. 4-1-2 = 1 last data card
	2-6	J	(I5)	the j^{th} element of i^{th} set
	7-10	blank		
	11-80	blank		
4-1-1-1	1	KEY	(I1)	= 0 continue to read data = 1 last data card
	2-6	K	(I5)	discretized time (1 to (II-1))
	7-11	blank		
	12-23	F(I,K,J)	(F12.6)	value of j^{th} element of i^{th} set forcing function at time K
	24-80	blank		repeat until all non-zero elements of F(I,K,J) are input

The whole deck would look like

4

4-1	4-2
4-1-1	4-2-1
4-1-1-1	4-2-1-1
4-1-1-2	4-2-1-2
4-1-1-3	4-2-1-3
⋮	⋮
4-1-2	4-2-2
4-1-2-1	4-2-2-1
4-1-2-2	4-2-2-2

Card No.	Columns	Variable	Format	Comments
				4-1-2-3 · 4-2-2-3
				⋮ ⋮ etc.

INITIAL CONDITIONS

5	1-17	INITIAL CONDITION**		only non-zero initial conditions are to be entered
	18-80	blank		
5- 1	1	KEY	(I1)	= 0 continue to read data cards = 1 last data card
	2-6	I	(I5)	the i^{th} element of the state vector $\bar{s}$
	7-10	blank		
	11-22	S (1,I)	(F 12.6)	the value of s(1,I)
	23-80	blank		continue with card 5-2, etc. until all non-zero elements of S(1,I) are input 5-2 has the same format as 5-1 The initial conditions are the values of the response (state) variables at the initial time

FIRST ORDER EQUATIONS OF MOTION ($\dot{\bar{s}} = \underline{A}\bar{s} + \underline{B}\bar{u} + \underline{D}\bar{f}_k$)

6	1-8	A MATRIX**		
	9-80	blank		
6-1	1	KEY	(I1)	= 0 continue to read data of $\underline{A}$ = 1 last data card of $\underline{A}$
	2-6	I	(I5)	

Card No.	Columns	Variable	Format	Comments
	7-11	J	(I5)	
	12-23	A(I,J)	(F12.6)	value of A(I,J)
	24-80	blank		repeat card 6-2,etc. until all non-zero elements of $\underline{A}$ are input
7	1-8	B MATRIX**		
	9-80	blank		
7-1	1	KEY	(I1)	= 0 continue to read data of $\underline{B}$
				= 1 last data card of $\underline{B}$
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	B(I,J)	(F12.6)	value of B(I,J)
	24-80	blank		repeat card 7-2, etc.,until all non-zero elements of $\underline{B}$ are input.
8	1-8	D MATRIX**		
	9-80	blank		
8-1	1	KEY	(I1)	= 0 continue to read data of $\underline{D}$
				= 1 last data card of $\underline{D}$
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	D(I,J)	(F12.6)	value of D(I,J)
	24-80	blank		repeat 8-2, etc. until all non-zero elements of $\underline{D}$ are input.

SECOND ORDER EQUATIONS OF MOTION ($\underline{M}\ddot{\bar{q}} + \underline{C}\dot{\bar{q}} + \underline{K}\bar{q} + \underline{U}\bar{u} = \underline{F}\bar{f}_k$)

Card No.	Columns	Variable	Format	Comments
9	1-8	M MATRIX**		
	9-80	blank		
9-1	1	KEY	(I1)	= 0 continue to read data of $\underline{M}$
				= 1 last data of $\underline{M}$

Card No.	Columns	Variable	Format	Comments
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	M(I,J)	(F12.6)	value of M(I,J)
	24-80	blank		repeat until all non-zero elements of <u>M</u> are input
10	1-8	<u>C MATRIX</u> **		
	9-80	blank		
10	1	KEY	(II)	= 0 continue to read data of <u>C</u>
				= 1 last data card of <u>C</u>
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	C(I,J)	(F12.6)	value of C(I,J)
	24-80	blank		repeat until all non-zero elements of <u>C</u> are input
11	1-8	<u>K MATRIX</u>**		
	9-80	blank		
11-1	1	KEY	(I1)	= 0 continue to read data of <u>K</u>
				= 1 last data card of <u>K</u>
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	K(I,J)	(F12.6)	value of K(I,J)
	24-80	blank		repeat until all non-zero elements of <u>K</u> are input
12	1-8	<u>F MATRIX</u>**		
	9-80	blank		
12-1	1	KEY	(I1)	= 0 continue to read data of <u>F</u>
				= 1 last data card of <u>F</u>
	2-6	I	(I5)	

Card No.	Columns	Variable	Format	Comments
	7-11	J	(I5)	
	12-23	F(I,J)	(F12.6)	value of F(I,J)
	24-80	blank		repeat until all non-zero elements of F are input
13	1-8	U MATRIX**		
	9-80	blank		
13-1	1	KEY	(I1)	= 0 continue to read data of U = 1 last data card of U
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	U(I,J)	(F12.6)	value of U(I,J)
	24-80	blank		repeat until all non-zero elements of U are input

OBJECTIVE FUNCTION $(\underline{PX1}\bar{s} + \underline{PX2}\bar{u} + \underline{PX3}\bar{f}_k)$

Card No.	Columns	Variable	Format	Comments
14	1-10	PX1 MATRIX**		
	11-80	blank		
14-1	1	KEY	(I1)	= 0 continue to read data of PX1 = 1 last data card of PX1

Card No.	Colunns	Variable	Format	Comments
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	PX1(I,J)	(F12.6)	value of PX1(I,J)
	24-80	blank		repeat until all non-zero elements of PX1 are input
15	1-10	PX2 MATRIX**		
	11-80	blank		
15-1	1	KEY	(I1)	= 0 continue to read data of PX2 = 1 last data card of PX2
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	PX2(I,J)	(F12.6)	value of PX2(I,J)
	24-80	blank		repeat until all non-zero elements of PX2 are input
16	1-10	PX3 MATRIX**		
	11-80	blank		
16-1	1	KEY	(I1)	= 0 continue to read data of PX3 = 1 last data of PX3
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	PX3(I,J)	(F12.6)	value of PX3(I,J)
	24-80	blank		repeat until all non-zero elements of PX3 are input

---

CONSTRAINTS ($\overline{YL} \leq \underline{Y1}\bar{s} + \underline{Y2}\bar{u} + \underline{Y3}\bar{f}_k \leq \overline{YU}$)

Card No.	Colunns	Variable	Format	Comments
17	1-9	Y1 MATRIX**		
	10-80	blank		

Card No.	Columns	Variable	Format	Comments
	1	KEY	(I1)	= 0 continue to read data of Y1 = 1 last data card of Y1
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	Y1(I,J)	(F12.6)	value of Y1(I,J)
	24-80	blank		repeat until all non-zero elements of Y1 are input
18	1-10	Y2 MATRIX**		
	11-80	blank		
18-1	1	KEY	(I1)	= 0 continue to read data of Y2 = 1 last data card of Y2
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	Y2(I,J)	(F12.6)	value of Y2(I,J)
	24-80	blank		repeat until all non-zero elements of Y2 are input
19	1-10	Y3 MATRIX**		
	11-80	blank		
19-1	1	KEY	(I1)	= 0 continue to read data of Y3 = 1 last data of Y3
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	Y3(I,J)	(F12.6)	value of Y3(I,J)
	24-80	blank		repeat until all non-zero elements of Y3 are input
20	1-21	BOUNDS OF CONSTRAINTS**		
	22-80	blank		

Card No.	Columns	Variable	Format	Comments
20-1	1	KEY1	(I1)	= 0 continue to read data card (20-2) = 1 last data card
	2-6	I	(I5)	the i^{th} constraint
	7-9	blank		
	10	MSP(I,1)	(I1)	= 0 i^{th} constraint is for all times = 1 i^{th} constraint is for specific time
	11-19	blank		
	20	MSP(I,2)	(I1)	= 0 ≦ ≧ type constraint = 1 ≦ type constraint = 2 ≧ type constraint = 3 = type constraint
	21-80	blank		

If MSP(I,1) = 0 and:

- MSP(I,2) = 0 GOTO 22-1AU
  then go to 22-1AL
- MSP(I,2) = 1 GOTO 22-1BU
- MSP(I,2) = 2 GOTO 22-1CL
- MSP(I,2) = 3 GOTO 22-1DU

If MSP(I,1) = 1 and:

- MSP(I,2) = 0 GOTO 22-1EU
  then go to 22-1EL
- MSP(I,2) = 1 GOTO 22-1FU
- MSP(I,2) = 2 GOTO 22-1GL
- MSP(I,2) = 3 GOTO 22-1HU

(Use the following if MSP(I,1) = 0, MSP(I,2) = 0)

22-1AU	1	blank		

Card No.	Columns	Variable	Format	Comments
	2	NCN	(I1)	= 0 constant upper bound (GO TO 22-1AU-A) = 1 piecewise constant upper bound (GO TO 22-1AU-1)
	3-80	blank		
22-1AU-A	1-12	SVA1	(F12.6)	starting value
	13-17	NIC1	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC1	(F12.6)	value of increment
	30-80	blank		(GO TO 22-1AL)
22-1AU-1	1	KEY	(I1)	= 0 continue to read data = 1 last data card
	2-6	K	(I5)	discretized time NOTE: K = 2 to II if Y2(I,J) = 0 for all J K = 1 to II - 1 if Y1(I,J) = 0 for all J K = 2 to II - 1 if Y1(I,J) ≠ 0 and Y2(I,J) ≠ 0 for some J
	7-11	blank		
	12-23	YU(K,1)	(F12.6)	
	24-80	blank		repeat 11-1AU-2, etc.
22-1AL	1	blank		
	2	NCN	(I1)	= 0 constant lower bound (GO TO 22-1AL-A) = 1 piecewise constant lower bound (GO TO 22-1AL-1)
	3-80	blank		
22-1AL-A	1-12	SVA2	(F12.6)	starting value
	13-17	NIC2	(I5)	number of increments (use 0 if there are no increments)

Card No.	Columns	Variable	Format	Comments
	18-29	VIC2	(F12.6)	value of increment
	30-80	blank		
22-1AL-1	1	KEY	(I1)	= 0 continue to read data = 1 last data card
	2-6	K	(I5)	discretized time NOTE: K = 2 to II if Y2(I,J) = 0 for all J K = 1 to II-1 if Y1(I,J) = 0 for all J K-2 to II-1 if Y1(I,J) $\neq$ 0 and Y2(I,J) $\neq$ 0 for some J
	7-11	blank		
	12-23	YL(K,I)	(F12.6)	
	24-80	blank		repeat 22-1AL-2, etc.

(Use the following if MPS(I,1) = 0 and MPS(I,2) = 1)

Card No.	Columns	Variable	Format	Comments
22-1BU	1	blank		
	2	NCN	(I1)	= 0 constant upper bound (GO TO 22-1BU-A) = 1 piecewise constant upper bound (GO TO 22-1BU-1)
	3-80	blank		
22-1BU-A	1-12	SVA1	(F12.6)	starting value
	13-17	NIC1	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC1	(F12.6)	value of increment
	30-80	blank		

Card No.	Columns	Variable	Format	Comments
22-1BU-1	1	KEY	(I1)	= 0 continue to read data = 1 last data card
	2-6	K	(I5)	discretized time NOTE: K = 2 to II if Y2(I,J) = 0 for all J K = 1 to II - 1 if Y1(I,J) = 0 for all J K = 2 to II-1 if Y1(I,J) $\neq$ 0 and Y2(I,J) $\neq$ 0 for some J
	7-11	blank		
	12-23	YU(K,I)	(F12.6)	
	24-80	blank		repeat 22-1BU-2, etc.

(Use the following if MSP(I,1) = 0, MSP(1,2) = 2)

Card No.	Columns	Variable	Format	Comments
22-1CL	1	blank		
	2	NCN	(I1)	= 0 constant lower bound (GO TO 22-1CL-A) = 1 piecewise constant lower bound (GO TO 22-1CL-1)
	3-80	blank		
22-1CL-A	1-12	SVA2	(F12.6)	starting value
	13-17	NIC2	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC2	(F12.6)	value of increment
	30-80	blank		
22-1CL-1	1	KEY	(I1)	= 0 continue to read data = 1 last data card
	2-6	K	(I5)	discretized time NOTE: K = 2 to II if Y2(I,J) = 0 for all J

Card No.	Columns	Variable	Format	Comments
				K = 1 to II-1 if Y1(I,J) = 0 for all J K = 2 to II-1 if Y1(I,J) ≠ 0 and Y2(I,J) ≠ 0 for some J
	7-11	blank		
	12-23	YL(K,I)	(F12.6)	
	24-80	blank		repeat 22-1CL-2, etc.

(Use the following if MSP(I,1) = 0, MSP(I,2) = 3)

Card No.	Columns	Variable	Format	Comments
22-IDU	1	blank		
	2	NCN	(I,1)	= 0 constant right hand side value (GO TO 22-1DU-A) = 1 piecewise constant right hand side value (GO TO 22-1DU-1)
	3-80	blank		
22-1DU-A	1-12	SVA1	(F12.6)	starting value
	13-17	NIC1	(I5)	number of movements (use 0 if there are no increments)
	18-29	VIC1	(F12.6)	value of increment
	30-80	blank		
22-1DU-1	1	KEY	(I1)	= 0 continue to read data = 1 last data card
	2-6	K	(I5)	discretized time NOTE: K = 2 to II if Y2(I,J) = 0 for all J K = 1 to II-1 if Y1(I,J) = 0 for all J K = 2 to II-1 if Y1(I,J) ≠ 0 and Y2(I,J) ≠ 0 for some J

Card No.	Columns	Variable	Format	Comments
	7-11	blank		
	12-23	YU(K,I)	(F12.6)	
	24-80	blank		repeat 22-1DU-2, etc.

(Use the following if MSP(I,1) = 1, MSP(I,2) = 0)

Card No.	Columns	Variable	Format	Comments
22-1EU	1	blank		
	2	NCN	(I1)	= 0 constant upper bound (GO TO 22-1EU-A)
	3-80	blank		
22-1EU-A	1-12	SVA1	(F12.6)	starting value
	13-17	NIC1	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC1	(F12.6)	value of increment
	30-34	K	(I5)	specify the time
	35-80	blank		
22-1EL	1	blank		
	2	NCN	(I1)	= 0 constant lower bound (GO TO 22-1EL-A)
	3-80	blank		
22-1EL-A	1-12	SVA2	(F12.6)	starting value
	13-17	NIC2	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC2	(F12.6)	value of increment
	30-34	K	(I5)	specify the time
	35-80	blank		

(Use the following if MSP(I,1) = 1,MSP(I,2) = 1)

Card No.	Columns	Variable	Format	Comments
22-1FU	1	blank		
	2	NCN	(I1)	= 0 constant upper bound (GO TO 22-1FU-A)
	3-80	blank		
22-1FU-A	1-12	SVA1	(F12.6)	starting value
	13-17	NIC1	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC1	(F12.6)	value of increment

Card No.	Columns	Variable	Format	Comments
	30-34	K(I)	(I5)	specify the time
	35-80	blank		

(Use the following if MSP(I,1) = 1, MSP(I,2) = 2)

22-1GL	1	blank		
	2	NCN	(I1)	= 0 constant lower bound (GO TO 22-1GL-A)
	3-80	blank		
22-1GL-A	1-12	SVA2	(F12.6)	starting value
	13-17	NIC2	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC2	(F12.6)	value of increment
	30-34	K	(I5)	specify the time
	35-80	blank		

(Use the following if MSP(I,L) = 1,MSP(I,2) = 3)

22-1HU	1	blank		
	2	NCN	(I1)	= 0 constant right-hand side (GO TO 22-1HU-A)
	3-80	blank		
22-1HU-A	1-12	SVA1	(F12.6)	starting value
	13-17	NIC1	(I5)	number of increments (use 0 if there are no increments)
	18-29	VIC1	(F12.6)	value of increment
	30-34	K	(I5)	
	35-80	blank		

OUTPUT ($\underline{Q1}\bar{s}$ + $\underline{Q2}\bar{u}$ + $\underline{Q3}\bar{f}_k$)

23	1-9	Q1 MATRIX**		
	10-80	blank		

Card No.	Columns	Variable	Format	Comments
23-1	1	KEY	(I1)	= 0 continue to read data
				= 1 last data card
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	Q1(I,J)		value of Q1(I,J)
	24-80	blank		repeat until all non-zero elements of Q1 are input
24	1-9	Q2 MATRIX**		
	10-80	blank		
24-1	1	KEY	(I1)	= 0 continue to read data
				= 1 last data card
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	Q2(I,J)	(F12.6)	value of Q2(I,J)
	24-80	blank		repeat until all non-zero elements of Q2 are input
25	1-9	Q3 MATRIX**		
	10-80	blank		
25-1	1	KEY	(I1)	= 0 continue to read data
				= 1 last data card
	2-6	I	(I5)	
	7-11	J	(I5)	
	12-23	Q3(I,J)	(F12.6)	value of Q3(I,J)
	24-80	blank		repeat until all non-zero elements of Q3 are input
26	1-10	TRAJECTORY**		
	11-80	blank		

Card No.	Columns	Variable	Format	Comments
26-1	1	KEY	(I1)	= 0 continue to read data e.g. 26-2
				= 1 last data card
	2-6	I	(I5)	the i^{th} output
	7-10	blank		
	11	ITR	(I1)	= 0 no plot wanted
				= 1 want to plot i^{th} output
	12-80	blank		
				repeat until all needed trajectories are specified
27	1-17	TRADE OFF DIAGRAM**		
	18-80	blank		
27-1	1-4	blank		
	5	ITD	(I1)	= 0 no plot wanted
				= 1 want tradeoff diagram
	6-9	blank		
	10	KSWT	(I1)	= 0 the ordinate of tradeoff curve starts from 0
				= 1 the ordinate of tradeoff curve starts from the minimum value of objective function
	11-80	blank		

VERIFICATION OF INPUT AND CALCULATED DATA

Card No.	Columns	Variable	Format	Comments
28	1-6	VERIFY**		
	7-80	blank		
28 - 1	1-4	blank		
	5	INP	(I1)	= 0 does not print input data
				= 1 print out input data

Card No.	Columns	Variable	Format	Comments
	6-9	blank		
	10	INT	(I1)	= 0 does not print data calculated by PREPROC = 1 print data calculated by PREPROC
	11-80	blank		

FINISH INPUT DATA

Card No.	Columns	Variable	Format	Comments
29	1-17	FINISH INPUT DATA*		
	18-80	blank		

MODIFICATION

Card No.	Columns	Variable	Format	Comments
30	1-14	MODIFY PROBLEM**		
	15-80	blank		Put the modified data between cards 30 and 31. The user can modify all but the data in the first three cards. The format of the modified data are the same as described before.
31	1-18	FINISH MODIFY DATA**		
	19-80	blank		

RESTARTING A NEW PROBLEM

Card No.	Columns	Variable	Format	Comments
32	1-7	RESTART**		
	8-80	blank		Put data from card 1 through card 31 after this card. This permits the user to run two or more different problems at a time.

END OF THE PROBLEM

Card No.	Columns	Variable	Format	Comments
33	1-4	STOP*		Use this card when there is no more data to be read
	5-80	blank		

APPENDIX II

LINEAR PROGRAMMING FORMULATION OF THE LIMITING PERFORMANCE PROBLEM

In this appendix the limiting performance problem is formulated as a linear programming (LP) problem. We consider dynamical systems in which those portions for which the performance is to be measured are represented by time dependent control forces. Although the control forces can replace nonlinear system segments, the remaining portions of the system must be linear. In addition, the kinematics of the whole system must be linear. Although the engineer frequently describes a dynamical system with second order differential equations, it is more convenient for us to work with first order equations. Hence, Section A of this appendix considers the conversion from second to first order equations.

The unknown control forces are discretized in time. Integration of the equations of motion for known disturbances yields the response (state) variables as linear combinations of the control forces. The objective functions and constraints comprising the limiting performance problem are formed from these response variables and hence are also linear functions of the discretized control forces. The limiting performance problem then can be formulated as a linear programming problem. Section B of this appendix treats the integration of the first order equations of motion containing the unknown control forces. Finally, we formulate the limiting performance problem as a linear programming problem in Section C.

A. Conversion of Equations of Motion of a Dynamic System to a System of First Order Equations

Consider the general equations of motion of a multi-degree-of freedom system in the form

$$\underline{M}\ddot{\bar{q}} + \underline{C}\dot{\bar{q}} + \underline{K}\bar{q} + \underline{U}\bar{u} = \underline{F}\bar{f} \qquad \text{(II-1)}$$

where

$\bar{q}$ = displacement vector (n x 1)

$\bar{u}$ = control force vector (nu x 1)

$\bar{f}$ = forcing function vector (nf x 1)

$\underline{M}$ = mass matrix (n x n)

$\underline{C}$ = damping matrix (n x n)

$\underline{K}$ = stiffness matrix (n x n)

$\underline{U}$ = matrix of constants associated with control forces (n x nu)

F = matrix of constants associated with external disturbances (n x nf)

Letters with bars over them refer to vectors, e.g.,

$$\bar{q} = \begin{bmatrix} q_1 \\ q_2 \\ \cdot \\ \cdot \\ \cdot \\ q_n \end{bmatrix} \text{ is the displacement vector,}$$

while letters with bars under them refer to matrices, e.g.,

$$\underline{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ c_{n1} & \cdots & \cdots & c_{nn} \end{bmatrix} \text{ is the damping coefficient matrix.}$$

These equations can be converted by the following procedure into a system of first order equations.

Let $\bar{s} = \left[\begin{array}{c} \dot{\bar{q}} \\ \hline \bar{q} \end{array}\right]$

where $\bar{s}$ is a partitioned vector. Also define

$$\underline{W} = \left[\begin{array}{c|c} \underline{O} & \underline{M} \\ \hline \underline{M} & \underline{C} \end{array}\right]_{2n\times 2n}$$

$$\underline{X} = \left[\begin{array}{c|c} -\underline{M} & \underline{O} \\ \hline \underline{O} & \underline{K} \end{array}\right]_{2n\times 2n}$$

$$\underline{Y} = \left[\begin{array}{c} \underline{O} \\ \hline \underline{U} \end{array}\right]_{2n\times nu}$$

$$\underline{Z} = \left[\begin{array}{c} O_2 \\ \hline \underline{F} \end{array}\right]_{2n\times nf}$$

Then the system of second order equations can be written as

$$\underline{W}\,\dot{\bar{s}} + \underline{X}\,\bar{s} + \underline{Y}\,\bar{u} = \underline{Z}\,\bar{f} \qquad \text{(II-2)}$$

or $\dot{s} = \underline{A}\,\bar{s} + \underline{B}\,\bar{u} + \underline{D}\,\bar{f}$

where

$$\underline{A} = \underline{W}^{-1}\,\underline{X} = \left[\begin{array}{c|c} -\underline{M}^{-1}\underline{C} & -\underline{M}^{-1}\,\underline{K} \\ \hline \underline{I} & \underline{O} \end{array}\right]_{2n\times 2n}$$

$$B = -W^{-1}\,Y = \left[\begin{array}{c} -\underline{M}^{-1}\,\underline{U} \\ \hline \underline{O}_1 \end{array}\right]_{2n\times nu}$$

$$D = W^{-1}\,Z = \left[\begin{array}{c} \underline{M}^{-1}\,\underline{F} \\ \hline \underline{O}_2 \end{array}\right]_{2n\times nf}$$

$\underline{O}$, $\underline{O}_1$ and $\underline{O}_2$ are null matrices of order nxn, nxnu and nxnf respectively.

$\underline{I}$ is an nxn identity matrix. Here we have been made use of

$$\underline{W}^{-1} = \left[\begin{array}{c|c} -\underline{M}^{-1}\ \underline{C}\ \underline{M}^{-1} & \underline{M}^{-1} \\ \hline \underline{M}^{-1} & \underline{O} \end{array}\right]_{2nx2n}$$

Note that $\underline{A}$, $\underline{B}$, $\underline{D}$, $\underline{W}$ and $\underline{W}^{-1}$ are partitioned matrices.

B. <u>Integration of a System of First Order Equations</u>

We wish to integrate the system of first order equations

$$\dot{\bar{s}} = \underline{A}\bar{s} + \underline{B}\bar{u} + \underline{D}\bar{f} \qquad \text{(II-3)}$$

with initial conditions

$$\bar{s}\big|_{t=0} = \bar{s}(o) = \left[\frac{\dot{\bar{q}}(o)}{\bar{q}(o)}\right]$$

The vectors $\bar{s}$, $\bar{u}$, $\bar{f}$, and matrices $\underline{A}$, $\underline{B}$, $\underline{D}$ are defined in the previous appendix.

To solve Eqs (II-3) let

$$\bar{s}(t) = e^{\underline{A}t}\ \bar{z}(t) \qquad \text{(II-4)}$$

Substitute Eq. (II-4) into Eqs. (II-3) to find

$$\underline{A}e^{\underline{A}t}\ \bar{z} + e^{\underline{A}t}\frac{d\bar{z}}{dt} = \underline{A}e^{\underline{A}t}\ \bar{z} + \underline{B}\bar{u} + \underline{D}\bar{f} \qquad \text{(II-5)}$$

If we substract $\underline{A}e^{\underline{A}t}\bar{z}$ from both sides of Eq (II-5) and then premultiply both sides by $e^{-\underline{A}t}$ we obtain

$$\frac{d\bar{z}}{dt} = e^{-\underline{A}t}\ [\underline{B}\bar{u} + \underline{D}\bar{f}]$$

and $\bar{z}(0) = \bar{s}(o)$, since $\bar{s}(o) = e^{\underline{A}(o)}\bar{z}(o) = \underline{I}\bar{z}(o) = \bar{z}(o)$.

Then,

$$\bar{z}(t) = \bar{s}(o) + \int_0^t e^{-\underline{A}y}[\underline{B}\bar{u}(y) + \underline{D}\bar{f}(y)]dy \qquad \text{(II-6)}$$

We see from Eqs. (II-4) and (II-6) that the solution of the system (II-3) can be written as

$$\bar{s}(t) = e^{\underline{A}t}[\bar{s}(o) + \int_0^t e^{-\underline{A}y}[\underline{B}\bar{u}(y) + \underline{D}\bar{f}(y)]dy] \qquad \text{(II-7)}$$

where y is a dummy variable for integration.

To complete the integration in Eq. (II-7) we must know $\bar{u}(t)$ and $\bar{f}(t)$. The loading functions $\bar{f}(t)$ are prescribed. The forces $\bar{u}(t)$ are unknowns to be found as solutions of an optimization problem. We discretize $\bar{u}(t)$ so that between two discretized times $\bar{u}$ is represented by constants. This will permit Eq. (II-7) to be integrated. For convenience, $\bar{f}(t)$ is discretized in the same fashion as $\bar{u}(t)$.

<u>Piecewise Constant Discretization</u>

Assume $\bar{u}$ is discretized as a piecewise constant function of time as shown in Fig. II-1. Here, between $t = iT$ and $t = (i + 1)T$, $\bar{u}(t) = \bar{u}(iT)$, where T is the time increment Δt.

Then, at any time kT

$$\bar{s}(kT) = e^{\underline{A}kT}[\bar{s}(o) + \int_0^{kT} e^{-\underline{A}y}[\underline{B}\bar{u}(y) + \underline{D}\bar{f}(y)]dy]$$

$$= e^{\underline{A}kT}\bar{s}(o) + e^{\underline{A}kT}\sum_{i=o}^{k-1} \int_{iT}^{(i+1)T} e^{-\underline{A}y}[\underline{B}\bar{u}(y) + \underline{D}\bar{f}(y)]dy$$

Now,

$$\int_{iT}^{(i+1)T} e^{-\underline{A}y}[\underline{B}\bar{u}(y) + \underline{D}\bar{f}(y)]dy$$

$$= \int_{iT}^{(i+1)T} e^{-\underline{A}y}[\underline{B}\bar{u}(iT) + \underline{D}\bar{f}(iT)]dy$$

$$= e^{-\underline{A}(i+1)T}A^{-1}[e^{\underline{A}T} - I][\underline{B}\bar{u}(iT) + \underline{D}\bar{f}(iT)]$$

Thus,

$$\bar{s}(kT) = e^{\underline{A}kT}\,\bar{s}(o) + \sum_{i=o}^{k-1} e^{\underline{A}(k-i-1)T}\underline{A}^{-1}(e^{\underline{A}T} - \underline{I})[\underline{B}\bar{u}(iT) + \underline{D}\bar{f}(iT)] \qquad \text{(II-8)}$$

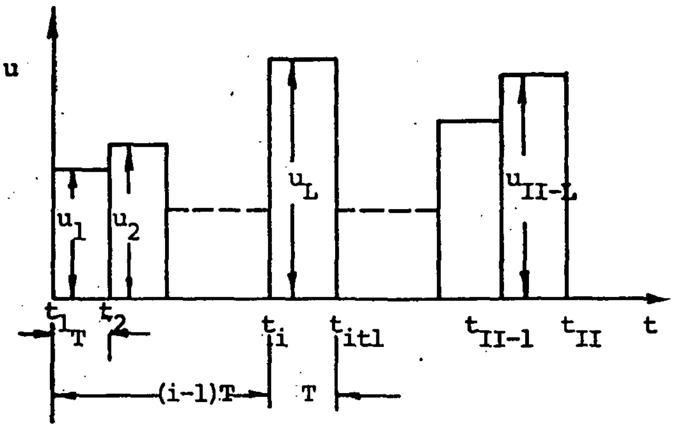


Fig. II-1 Piecewise Constant Discretization of u (u is any Control Force Function).

is the required discrete formal integration of Eq. (II-3), for a piecewise constant representation of $\bar{u}(t)$. Note, here

$$\underline{A}^{-1}(e^{\underline{A}T} - \underline{I}) = \underline{A}^{-1}\left(\underline{I} + \underline{A}T + \frac{\underline{A}^2T^2}{2!} + \frac{\underline{A}^3T^3}{3!} + \cdots - \underline{I}\right)$$

$$= \underline{I}T + \frac{\underline{A}T^2}{2!} + \frac{\underline{A}^2T^3}{3!} + \frac{\underline{A}^3T^4}{4!} + \cdots \qquad \text{(II-9)}$$

Even if $\underline{A}$ is singular, the solution, i.e., Eq. (II-8), is still valid since the series on the right-hand side of Eq. (II-9) always exists.

C. Formulation of the Linear Programming Problem

In this Section the limiting performance problem is formulated as a linear programming (LP) problem, with discretized control forces as unknowns. This is accomplished using the objective functions and constraints in the form described in Appendix I together with the solution of the equations of motion given in Section B of this Appendix.

The solution of the first order equations of motion are given in Section B for the piecewise constant discretization of the control forces as:

$$\bar{s}(kT) = e^{\underline{A}kT}\bar{s}(o) + \sum_{i=o}^{k-1} e^{\underline{A}(k-i-1)T}\,\underline{A}^{-1}(e^{\underline{A}T}-\underline{I})\,[\underline{B}\,\bar{u}(iT) + \underline{D}\bar{f}\,(iT)] \qquad \text{(II-10)}$$

In order to be compatible with indices in FØRTRAN, let $\bar{s}(1)$ be the initial condition $\bar{s}(0)$. We rewrite equation (II-10) as:

$$\bar{s}(k) = (e^{\underline{A}T})^{k-1}\bar{s}(1) + \sum_{i=1}^{k-1} (e^{\underline{A}T})^{k-1-i}\underline{A}^{-1}(e^{\underline{A}T} - \underline{I})[\underline{B}\bar{u}(i) + \underline{D}\bar{f}\,(i)] \qquad \text{(II-11)}$$

where $\bar{s}(1) = \bar{s}(0)$

$\bar{s}(k) = \bar{s}((k-1)T)$

Now, define the following matrices:

$$\begin{aligned}
\underline{P} &= e^{\underline{A}T} \\
\underline{AEI} &= \underline{A}^{-1}(e^{\underline{A}T}-\underline{I}) \\
\underline{R}(n) &= \underline{P}^{n-1}\underline{AEI}\ \underline{B} \\
\underline{T}(n) &= \underline{P}^{n-1}\underline{AEI}\ \underline{D}
\end{aligned} \qquad \text{(II-12)}$$

Note that $\underline{R}(1) = \underline{AEI}\ \underline{B}$, $\underline{T}(1) = \underline{AEI}\ \underline{D}$

Here $\underline{R}(i)$ and $\underline{T}(i)$ $(i = 1,2,\ldots n)$ are matrices of order m x nu, m x nf, respectively, where m = number of first order equations to be integrated and n is the number of discretized time intervals. In terms of those new matrices Eq. (II-11) becomes

$$\bar{s}(k) = \underline{P}^{k-1}\bar{s}(1) + \sum_{i=1}^{k-1}[\underline{R}(k-i)\bar{u}(i) + \underline{T}(k-i)\bar{f}(i)] \qquad \text{(II-13)}$$

Now, define

$$\begin{aligned}
\underline{RG}(k) &= [\underline{R}(k-1) \vdots \underline{R}(k-2) \vdots \cdots \vdots \underline{R}(1)] \\
\underline{TG}(k) &= [\underline{T}(k-1) \vdots \underline{T}(k-2) \vdots \cdots \vdots \underline{T}(1)]
\end{aligned}$$

$$\bar{\bar{f}}(k) = \begin{bmatrix} \bar{f}(1) \\ \bar{f}(2) \\ \vdots \\ \bar{f}(k-1) \end{bmatrix} \qquad \text{(II-14)}$$

$$\bar{\bar{u}}(k) = \begin{bmatrix} \bar{u}(1) \\ \cdots \\ \bar{u}(2) \\ \cdots \\ \vdots \\ \cdots \\ \bar{u}(k-1) \end{bmatrix}$$

Using Eqs. (II-14), Eq. (II-13) becomes

$$\bar{s}(k) = \underline{P}^{k-1}\bar{s}(1) + \underline{RG}(k)\bar{\bar{u}}(k) + \underline{TG}(k)\bar{\bar{f}}(k) \qquad \text{(II-15)}$$

Let

$$\overline{C}(k) = \overline{C1}(k) + \overline{C2}(k) \tag{II-16}$$

where $\overline{C1}(k) = \underline{P}^{k-1}\overline{s}(1)$ (II-17)

$$\overline{C2}(k) = \underline{TG}(k)\overline{f}(k)$$

We can simplify Eq. (II-15) to

$$\overline{s}(k) = \overline{C}(k) + \underline{RG}(k)\overline{\overline{u}}(k) \tag{II-18}$$

For a system subject to several sets of forcing functions, Eq. (II-18) becomes

$$\overline{s}(s,k) = \overline{C}(s,k) + RG(k)\overline{\overline{u}}(k) \tag{II-19}$$

where

$$\overline{C}(s,k) = \overline{C1}(k) + \overline{C2}(s,k)$$

$\overline{C1}(k)$ is given in Eq (II-17)

$$\overline{C2}(s,k) = \underline{TG}(k)\overline{\overline{f}}(s,k)$$

$$\overline{\overline{f}}(s,k) = \begin{bmatrix} \overline{f}(s,1) \\ \cdots\cdots \\ \overline{f}(s,2) \\ \vdots \\ \cdots\cdots \\ \overline{f}(s,k-1) \end{bmatrix}$$

Eq. (II-19) expresses the state variable vector $\overline{s}$ as a linear function of the control forces, initial conditions, and forcing functions. Here the initial conditions and forcing functions are prescribed, the only unknowns are the control forces (u).

Since the objective functions and constraints are all formed as linear combinations of $\overline{s}$, $\overline{u}$, $\overline{f}$, they are all linear functions of the u's. Then the problem of finding the u's that minimize the maximum or maximize the minimum value of the objective functions subject to constraints becomes a problem in linear programming. We will now define the limiting performance

problem in terms of the LP problem and explain how to set up the data in the proper form.

The LP problem is to find $\bar{z}$ such that $\bar{c}\ \bar{z}$ is minimized (or maximized) subject to $\underline{H}\bar{z} \lesseqgtr \bar{G}$ (II-20)

Where $\bar{c} = (1, 0, 0, \ldots\ldots, 0)$ is a row vector

$$\bar{z} = \begin{bmatrix} \phi \\ \bar{\bar{u}}\ (II) \end{bmatrix}$$

$$\text{with } \bar{\bar{u}}\ (II) = \begin{bmatrix} \bar{u}(1) \\ \bar{u}(2) \\ \vdots \\ \bar{u}(II-1) \end{bmatrix} \qquad \text{(II-21)}$$

$\underline{H}$ and $\bar{G}$ are a matrix and a vector constructed from the objective functions and constraints, respectively.

The objective function is formed as

$$\bar{\Psi} = \underline{PX1}\bar{s} + \underline{PX2}\bar{u} + \underline{PX3}\bar{f}(s) \qquad \text{(II-22)}$$

where the index s designates the s^{th} set of forcing functions are being used. There are two cases of interest: minimizing the maximum value of $\bar{\Psi}$ or maximizing the minimum value. The minimization of the maximum value of $\bar{\Psi}$ is equivalent to finding the minimum of ϕ such that

$$|\Psi_i| \leq \phi \text{ for all } i = 1 \text{ to NOB}$$

$$\text{or} \quad -\phi + \Psi_i \leq 0 \qquad \text{(II-23)}$$

$$\phi + \Psi_i \geqslant 0 \qquad \text{(II-24)}$$

where Ψ_i is the i^{th} row of $\bar{\Psi}$

Using Eqs. (II-19), (II-22), Eqs. (II-23),(II-24) become

$$-\phi + \underline{PX1}_i\ \underline{RG}(k)\bar{\bar{u}}(k) + \underline{PX2}_i\bar{u}(k) \leq -\underline{PX1}_i\bar{C}(s,k) - \underline{PX3}_i\bar{f}(s,k) \qquad \text{(II-25)}$$

$$\phi + \underline{PX1}_i\underline{RG}(k)\bar{\bar{u}}(k) + \underline{PX2}_i\bar{u}(k) \geq -\underline{PX1}_i\bar{C}(s,k) - \underline{PX3}_i\bar{f}(s,k) \qquad \text{(II-26)}$$

Eqs. (II-25), (II-26) are valid for k = I1, I1 + 1, ... L

s = 1,2, ... NSETS

where $\underline{PX1}_i$, $\underline{PX2}_i$, $\underline{PX3}_i$ are the i^{th} rows of the matrices $\underline{PX1}$, $\underline{PX2}$, $\underline{PX3}$ respectively. The values of I1 and L depend on whether $\underline{PX1}_i$ or $\underline{PX2}_i$ are zero. If $\underline{PX1}_i = (0\ 0\ \ldots\ 0)$ then I1 = 1, L = II-1. If $\underline{PX2}_i = (0\ 0\ \ldots\ 0)$, then I1 = 2, L = II. If $\underline{PX1}_i \neq (0\ 0\ \ldots\ 0)$ and $\underline{PX2}_i \neq (0\ 0\ \ldots\ 0)$, then I1 = 2, L = II-1. The contribution to $\underline{H}$ and $\overline{G}$ for a typical objective function (e.g., i^{th} row of $\overline{\Psi}$) can be found from Eqs. (II-25) and (II-26). It is put in matrix form in Table II-1. The total contribution to $\underline{H}$ and $\overline{G}$ from the objective function is a matrix containing NOB submatrices in the form of Table (II-1), where NOB is the number of objective functions.

To maximize the minimum value of $\overline{\Psi}$ is equivalent to finding the maximum of ϕ such that $|\Psi i| \geq \phi$. Note that to find maximum ϕ is equivalent to finding minimum $-\phi$. Thus the LP problem for this case is still in the form of Eq. (II-20) except now the row vector $\overline{c}$ is (-1,0,0, ... 0). The formation of arrays $\underline{H}$ and $\overline{G}$ for the objective functions remain the same as above except the inequality signs are reversed.

The constraints are formed as

$$\underline{Y1}\,\overline{s} + \underline{Y2}\,\overline{u} + \underline{Y3}\,\overline{f}(s)$$

The type of i^{th} constraint depends on the value of the i^{th} row of the specification matrix, i.e. MSP(i,1) and MSP(i,2). The specification matrix has been defined in Appendix I. For MSP(i,1) = 0, the constraint is applied at all times.

1. MSP(i,2) = 0

The constraint is

$$\overline{YL}_i \leq \underline{Y1}_i\,\overline{s} + \underline{Y2}_i\,\overline{u} + \underline{Y3}_i\,\overline{f}(s) \leq \overline{YU}_i \qquad \text{(II-27)}$$

$$
\begin{bmatrix}
-1 & \underline{D}_{i1} & \underline{E}_i & \underline{0} & & \cdot & \cdot & \cdot & \underline{0} \\
-1 & \underline{D}_{i1} & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \cdot & \cdot & \cdot & \underline{0} \\
\vdots & & & & & & & & \vdots \\
-1 & \underline{D}_{i(k-1)} & \underline{D}_{i(k-2)} & \cdots & & \underline{D}_{i1} & \underline{E}_i & \underline{0} \cdot & \underline{0} \\
\vdots & & & & & & & & \vdots \\
-1 & \underline{D}_{i(II-1)} & \underline{D}_{i(II-2)} & \cdots & & \underline{D}_{i(k-1)} & \cdots & \underline{D}_{i1} & \underline{E}_i
\end{bmatrix}
\begin{bmatrix}
\phi \\ \bar{u}(1) \\ \bar{u}(2) \\ \vdots \\ \bar{u}(k-1) \\ \vdots \\ \bar{u}(II-1)
\end{bmatrix}
\begin{matrix}
\le \\ \le \\ \\ \le \\ \\ \\ \le
\end{matrix}
\begin{bmatrix}
B_1 \\ B_2 \\ \vdots \\ B_k \\ \cdots \\ \vdots \\ B_{II-1}
\end{bmatrix}
$$

$$
\begin{bmatrix}
1 & \underline{D}_{i1} & \underline{E}_i & \underline{0} & & \cdot & \cdot & \cdot & \underline{0} \\
1 & \underline{D}_{i2} & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \cdot & \cdot & \cdot & \underline{0} \\
\vdots & & & & & & & & \vdots \\
1 & \underline{D}_{i(k-1)} & \underline{D}_{i(k-2)} & & & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \underline{0} \\
\vdots & & & & & & & & \vdots \\
1 & \underline{D}_{i(II-1)} & \underline{D}_{i(II-2)} & \cdots & & \underline{D}_{i(k-1)} & \cdots & \underline{D}_{i1} & \underline{E}_i
\end{bmatrix}
\begin{bmatrix}
\phi \\ \bar{u}(1) \\ \bar{u}(2) \\ \vdots \\ \bar{u}(k-1) \\ \vdots \\ \bar{u}(II-1)
\end{bmatrix}
\begin{matrix}
\ge \\ \ge \\ \\ \ge \\ \\ \\ \ge
\end{matrix}
\begin{bmatrix}
B_1 \\ B_2 \\ \vdots \\ B_k \\ \vdots \\ B_{II-1}
\end{bmatrix}
$$

where

$$\underline{D}_{ij} = \underline{PX1}_i \; \underline{R}(j)$$

$$\underline{E}_i = \underline{PX2}_i$$

$$B_k = -\underline{PX1}_i \; \bar{C}(s,k+1) - \underline{PX3}_i \, \bar{f}(s,k+1)$$

TABLE II-1

Using Eq. (II-19), Eq. (II-27) can be written as

$$\underline{Y1}_i\underline{RG}(k)\bar{\bar{u}}(k) + \underline{Y2}_i\bar{u}(k) \leq \overline{YU}_i(k) - \underline{Y1}_i\bar{C}(s,k) - \underline{Y3}_i\bar{f}(s,k) \tag{II-28}$$

$$\underline{Y1}_i\underline{RG}(k)\bar{\bar{u}}(k) + \underline{Y2}_i\bar{u}(k) \geq \overline{YL}_i(k) - \underline{Y1}_i\bar{C}(s,k) - \underline{Y3}_i\bar{f}(s,k)$$
$$\text{for } K = I1 \text{ to } L \tag{II-29}$$

The contribution to $\underline{H}$ and $\bar{G}$ of Eqs. (II-28), (II-29) are summarized in Table II-2A, and Table II-2B, respectively. The value of I1 and L depends on whether $\underline{Y1}_i$ or $\underline{Y2}_i$ is zero. If $Y1_i = (0\ 0\ \ldots\ 0)$, then $I1 = 1$, $L = II-1$. If $\underline{Y2}_i = (0\ 0\ \ldots 0)$, then $I1 = 2$, $L = II$. If $\underline{Y1}_i \neq (0\ 0\ \ldots\ 0)$ and $\underline{Y2}_i \neq (0\ 0\ \ldots\ 0)$, then $I1 = 2$, $L = II-1$.

2. MSP(i,2) = 1
   The i^{th} constraint is

$$\underline{Y1}_i\bar{s} + \underline{Y2}_i\bar{u} + \underline{Y3}_i\bar{f}(s) \leq \overline{YU}_i \tag{II-30}$$

The contribution to $\underline{H}$ and $\bar{G}$ is the same as Eq. (II-28).

3. MSP(i,2) = 2

   Here the constraint is expressed as

$$\underline{Y1}_i\bar{s} + \underline{Y2}_i\bar{u} + \underline{Y3}_i\bar{f}(s) \geq \overline{YL}_i \tag{II-31}$$

The contribution to $\underline{H}$ and $\bar{G}$ is the same as Eq. (II-29).

4. MSP(i,2) = 3

   In this case the constraint is given by

$$\underline{Y1}_i\bar{s} + \underline{Y2}_i\bar{u} + \underline{Y3}_i\bar{f}(s) = \overline{YU}_i \tag{II-32}$$

The contribution to $\underline{H}$ and $\bar{G}$ is the same as Eq. (II-28), except now the inequality becomes an equality. For MSP(i,1) = 1, the constraint is applied at a specified time only. The formal contribution to $\underline{H}$ and $\bar{G}$ for all cases is the same as for MSP(i,1) = 0, except k is now restricted to a specific time, i.e. , KS(i).

$$
\begin{bmatrix}
0 & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \cdot & \cdot & \cdot & & & \underline{0} \\
0 & \underline{D}_{i2} & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \cdot & \cdot & \cdot & & \underline{0} \\
 & \vdots & & & & & & & & \vdots \\
0 & \underline{D}_{i(k-1)} & \underline{D}_{i(k-2)} & \cdots & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \cdots & & \underline{0} \\
 & \vdots & & & & & & & & \vdots \\
0 & \underline{D}_{i(II-1)} & \underline{D}_{i(II-2)} & \cdots & \underline{D}_{i(k-1)} & \cdots & & \underline{D}_{i1} & & \underline{E}_i
\end{bmatrix}
\begin{bmatrix}
\phi \\ \bar{u}(1) \\ \bar{u}(2) \\ \vdots \\ \bar{u}(k-1) \\ \vdots \\ \bar{u}(II-1)
\end{bmatrix}
\begin{matrix}
\le \\ \le \\ \\ \le \\ \\ \\ \le
\end{matrix}
\begin{bmatrix}
B_1 \\ B_2 \\ \vdots \\ B_k \\ \vdots \\ B_{II-1}
\end{bmatrix}
$$

Where

$$\underline{D}_{ij} = \underline{Y1}_i \underline{R}(j)$$

$$\underline{E}_i = \underline{Y2}_i$$

$$B_k = \overline{YU}_i(k+1) - \underline{Y1}_i \overline{C}(s,k+1) - \underline{Y3}_i \bar{f}(s,k+1)$$

TABLE II-2A

$$
\begin{bmatrix}
0 & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \cdot & \cdot & \cdot & & & \underline{0} \\
0 & \underline{D}_{i2} & \underline{D}_{i1} & \underline{E}_i & \underline{0} & & \cdot & \cdot & \cdot & \underline{0} \\
 & \vdots & & & & & & & & \vdots \\
0 & \underline{D}_{i(k-1)} & \underline{D}_{i(k-2)} & \cdots & \underline{D}_{i1} & \underline{E}_i & \underline{0} & \cdot & \cdot & \underline{0} \\
 & \vdots & & & & & & & & \vdots \\
0 & \underline{D}_{i(II-1)} & \underline{D}_{i(II-2)} & \cdots & \underline{D}_{i(k-1)} & \cdots & & \underline{D}_{i1} & & \underline{E}_i
\end{bmatrix}
\begin{bmatrix}
\phi \\ \bar{u}(1) \\ \bar{u}(2) \\ \vdots \\ \bar{u}(k-1) \\ \vdots \\ \bar{u}(II-1)
\end{bmatrix}
\begin{matrix}
\geq \\ \geq \\ \\ \geq \\ \\ \\ \geq
\end{matrix}
\begin{bmatrix}
B_1 \\ B_2 \\ \vdots \\ B_k \\ \vdots \\ B_{II-1}
\end{bmatrix}
$$

where

$$\underline{D}_{ij} = \underline{Y1}_i R(j)$$

$$\underline{E}_i = \underline{Y2}_i$$

$$B_k = \overline{Y}L_i(k+1) - \underline{Y1}_i \overline{C}(s,k+1) - \underline{Y3}_i \overline{f}(s,k+1)$$

TABLE II-2B

D. Dimension of the Linear Programming Problem

The dimension of the linear programming problem depends on the number of controllers of the system, the number and nature of objective functions and constraints, the number of time intervals, and the number of sets of forcing functions. It is independent of the number of degrees of freedom of the system. The dimensions of the matrix $\underline{H}$ and vector $\overline{G}$ can be calculated as following:

$$R_H = [(II-1) \times (2N_1 + 2C_1 + C_3) + (II-2) \times (2N_2 + 2C_2 + C_4) + 2C_5 + C_6] \times NSETS$$

$$C_H = 1 + Nu \times (II-1)$$

$$R_G = R_H$$

where

R_H = Number of rows of matrix $\underline{H}$

C_H = Number of columns of matrix $\underline{H}$

R_G = Number of rows of vecter $\overline{G}$

N_1 = Number of objective functions for which $\underline{PX1_i} = 0$ or $\underline{PX2_i} = 0$,

N_2 = Number of objective functions for which $\underline{PX1_i} \neq 0$ and $\underline{PX2_i} \neq 0$

C_1 = Number of constraints for which MSP(I,1) = 0, MSP(I,2) = 0 and $\underline{Y1_i} = 0$ or $\underline{Y2_i} = 0$

C_2 = Number of constraints for which
MSP(I,1) = 0,MSP(I,2) = 0 and $\underline{Y1}_i \neq 0$ and $\underline{Y2}_i \neq 0$

C_3 = Number of constraints for which
MSP(I,1) = 0, MSP(I-2) $\neq$ 0 and $\underline{Y2}_i = 0$ or $\underline{Y1}_i = 0$

C_4 = Number of constraints for which
MSP(I,1) = 0,MSP(I,2) $\neq$ 0 and $\underline{Y1}_i \neq 0$ and $\underline{Y2}_i \neq 0$

C_5 = Number of constraints with MSP(I,1) = 1,MSP(I,2) = 0

C_6 = Number of constraints with MSP(I,1) = 1,MSP(I,2) $\neq$ 0

II = Number of time intervals used (see Appendix I)

NSETS = Number of sets of forcing functions

As an example, consider the dimensions of the linear programming problem for the SDF problem shown on page 17. Here II=10, N_2 = 1, C_1 = 1, NSETS = 1, Nu = 1

R_H = [(10-1) x (2 x 1 + 2 x 1)] x 1 = 36

C_H = 1 + 1 x (10-1) = 10.

APPENDIX III

PROGRAMMING DESCRIPTION OF THE COMPONENTS OF PERFORM

This appendix contains descriptions of the three major components of PERFORM: PREPROC, MPS/360, PSTPROC.

A. PREPROC

This section includes a description of program PREPROC. Appendix IV A contains a listing of program PREPROC.

Table III-1 contains a listing of the arrays used in the program with explanatory notes.

The variables beginning with an IVE card followed by a number are used to communicate to the printing routine those arrays which contain nonzero elements or special conditions. Table III-2 relates IVE numbers to program arrays.

1. Main Section

The main section of the program is described in the PREPROC section of Section III of this report.

2. Subroutines (Alphabetical Order)

a. BODATA

This subroutine punches the BOUNDS action of the MPS/360 linear programming problem. The default bounds for MPS/360 variables are zero lower and unlimited upper; because solutions to limiting performance problems can be negative, the default overriding BOUNDS section is necessary.

b. BRIDGE

This subroutine calculates the arrays $\underline{R}$, $\underline{T}$, $\overline{C1}$, $\overline{C2}$, and $\overline{C}$ as described in Appendix II. It is called from the main section.

Array Identifiers	Dimension	Notes	Common Bloc
A	m* x m	First order system A is input. Second order A computed by subroutine COVERT	ABD
B	m* x nu	First order B is input. Second order computed by subroutine COVERT	ABD
CAY	n x n	Matrix K in model	EKC
CC	n x n	Matrix C in model	EKC
CM	n x n	Becomes $\underline{M}^{-1}\underline{C}$ in subroutine COVERT.	INV
CYM	n x n	Becomes $\underline{M}^{-1}\underline{K}$ in subroutine COVERT.	INV
D	m* x nf	First order $\underline{D}$ is input. Second order computed by COVERT.	ABD
EM	n x n	Matrix M in model	EKC
EMI	n x n	Becomes $\underline{M}^{-1}$ in subroutine MIV.	INV
F	nsets x II x nf	Subscript 1: forcing function set number, 2: time interval, 3: row of equations of motion with which associated. This is not model matrix $\underline{F}$.	FOFUN
FF	n x nf	Matrix $\underline{F}$ in model.	EKC
FM	n x nf	Becomes $\underline{M}^{-1}\underline{F}$ in subroutine COVERT	EKC
G	number of LP rows (See Dimension of LP problem)	Right-hand side of LP tableau	RHV
H	number of LP rows; number of LP columns (See Dimension of LP problem)	Linear programming tableau.	PVAL

TABLE III-1. PREPROC ARRAYS

Array Identifiers	Dimension	Notes	Common Block
ISET	not	ISET(1) identifies forcing function of the set to be used in tabulation row I	TAB
ITITL		Alphanumeric problem title	TIT
ITR	not	ITR(I) is 0 if plot of I^{th} output is not desired. Otherwise plot is made	
IXX	noc	IXX(I) indicates the behavior of the bounds of I^{th} constraint = 0 no varying bounds = 1 $\overline{YU}$ varies = 2 $\overline{YL}$ varies = 3 $\overline{YL}$ and $\overline{YU}$ vary	
KS	noc	KS(I) specifies the specific time for I^{th} constraints	
MSP	noc x 2	Specific type of constraint See Appendix I for description.	ISP
NRE	number of LP rows	Identifies relation of each row with its right-hand side "G" means $\geq$ "L" means $\leq$ "E" means =	RHV
PX1	nob x m	* *	OBJ
PX2	nob x nu	* *	OBJ
PX3	nob x nf	* *	OBJ
Q1	not x m	* * *	TAB
Q2	not x nu	* * *	TAB
Q3	not x nf	* * *	TAB
S	2n	Used for initial condition in program	INIT
UM	n x nu	Becomes $\underline{M}^{-1}\underline{U}$ in subroutine COVERT	INV
UU	n x n	Matrix U in model	EKC

TABLE III-1. PREPROC Arrays (continued).

Array Identifiers	Dimension	Notes	Common Blo[ck]
Y	member of varying bounds	Array of varying bounds PSTPROC tradeoff diagram	UPLO
Y1	noc x m	* * * *	CONST
Y2	noc x nu	* * * *	CONST
Y3	noc x nf	* * * *	CONST
YU	II x noc	Upper bound vector. Subscript 1: time interval 2: number of constraints	UPLO
YL	II x noc	Lower bound vector, subscripts have same meaning as above.	UPLO
C	nsets x II x m	Arrays pertinent to the formation of right-hand side of the LP problem. (See Appendix II). Subscript 1: forcing function set 2: time 3: row number in state (response) vector	CCS
P	m x m	Becomes $e^{\underline{A}T}$ in BRIDGE	
C2		Arrays used in form array C.	
C1	II x m	Arrays used to form array C.	
R	II x m x m	Arrays pertinent to expressing responses as function of the control forces.	BRG
T	II x m x nf	Arrays pertinent to expressing responses as functions of forcing functions.	BRG
AEI	m x m	Becomes $-\underline{A}^{-1}(e^{\underline{A}t} - \underline{I})$ in BRIDGE	MYAP

TABLE III-1. PREPROC Arrays (continued)

Symbols used in array table.

nob = Number of objective functions (rows in PX1, PX2, PX3 expression).

noc = Number of constraint expressions.

nsets =Number of sets of forcing functions.

not = Number of tabulation expressions.

nf = Number of forcing functions.

nu = Number of isolators (control forces).

* m = NDF For first order system. m = 2(NDF) for second order system.

** Arrays related to input of objective functions, see Appendix I.

*** Arrays related to input of required output, see Appendix I.

**** Arrays related to input of constraints, see Appendix I.

NDF Number of equations in the system of equations.

TABLE III-1. PREPROC Arrays (concluded)

IVE	Array Read or Command
1	A
2	B
3	C
4	EM
5	CL
6	CAY
7	FF
8	UU
9	S (initial condition)
10	F (forcing function)
11	PX1
12	PX2
13	PX3
14	Y1
15	Y2
16	Y3
17	MSP
18	(Not used)
19	Q1
20	Q2
21	Q3

TABLE III-2. IVE Variable Array Relation

c. CODATA

This section punches the COLUMNS section of the MPS/360 file. The coefficients of the tableau array $\underline{H}$ are in this section, CODATA is called by LPDATA which is called by the main section.

d. COVERT

This subroutine reduces the equations of a second order system to those of a first order system. It is called from the main section.

e. ELEMTP

ELEMTP is called from the main section to compute the objective function portions of the $\underline{H}$ and $\overline{G}$ arrays.

f. ELEMTQ

The constraint portions of the $\underline{H}$ and $\overline{G}$ arrays are computed by subroutines called by ELEMTQ which is called from the main section. The routines called depend on whether the programming tableau rows are equalities or inequalities and whether $\overline{YL}$, $\overline{YU}$, or both are part of problem constraints.

g. ELEMTR

Subroutine ELEMTR computes the appropriate portions of the $\underline{H}$ and $\overline{G}$ array for the problem constraint I which is a passed parameter from ELEMTQ. Table II-2A of Appendix II shows the computation of ELEMTR. As can be noted, this routine computes only the $\overline{YU}$ array.

h. ELEMTS

This subroutine completes portions of the $\underline{H}$ and $\overline{G}$ arrays dependent on problem constraint I which is a passed parameter of ELEMTQ. Table II-2B of Appendix II shows the computation of ELEMTS. Note that this routine works only with $\overline{YL}$.

i. ELEMTT

Routine ELEMTT is called when the constraint of passed parameter I is an equality. Note that equalities always involve the $\overline{YU}$ vector.

j. LPDATA

The subroutine LPDATA punches section header cards for the MPS/360 linear programming problem and calls other subroutines to punch the actual section records. Subroutines RODATA, RHDATA, BODATA, and CODATA are called. LPDATA is called from the main section.

k. M12X3

This subroutine multiplies a two-dimensional array times a portion of a three-dimensional one, both of which are parameters.

l. M22X2

This routine multiplies two passed arrays.

m. MEXAP

This subroutine evaluates the matrices P and AEI used in Section C of Appendix III. These matrix functions are computed using approximate formulas described in Ref. 9. This routine can be replaced by any routine that uses other algorithms to compute matrix exponential functions.

n. MIV

This routine computes the inverse of array EM and stores it in array EMI.

o. POST

Subroutine POST punches the PSTPROC report specifications. It includes arrays Q1, Q2, Q3, Y, R, F, C, ISET and ITC. No subroutines are called by POST.

p. PRNT

After the linear programming problem has been compiled subroutine PRNT is called to print the H, G, and NRE arrays. Array NRE shows the sign (≤, = , ≥) of the linear programming row.

q. RHDATA

Subroutine RHDATA is called by LPDATA to punch the RHS section of the MPS/360 linear programming problem.

r. RODATA

Subroutine RODATA punches the ROWS section of the MPS/360 linear programming problem. The ROWS section associates names with the tableau rows and specifies the relation of each row to its right-hand side, i.e., $\leqslant$, $=$, or $\geqslant$.

s. SEARCH

This subroutine sets a parameter to 1 if there are nonzero elements in the given row of the passed two-dimensional array. Otherwise a zero is returned.

t. VERFY

If a VERIFY indicator record is encountered and variable INP is not zero, subroutine VERFY is called to print all of the PERFORM problem specification data.

u. VERFY2

If variable INT is nonzero, subroutine VERFY2 is called to print out the intermediate resulting arrays. These include $\underline{A}$, $\underline{B}$, $\underline{D}$, $\underline{R}$, $\underline{AEI}$.

B. MPS/360

A listing and discussion of the MPS/360 control program is in this section. The material about MPS and its relation to PERFORM in part B of the Section III of this report is useful background information to this appendix.

Consider the listing of the MPS/360 control program. The first two statements, PROGRAM and TITLE, identify the program name and the title to be printed on each page of the solution report.

The first instruction is the macro INITIALZ. Macro instructions direct the compiler to include at the point of appearance a predefined set of source instructions identified by the macro name; macros eliminate the need for tedious programmer coding of repeatedly used sets of instructions. The INITIALZ macros initializes values of certain tolerances, standard frequencies, etc, and specifies actions to be taken should errors occur during execution.

MPS/360 Control Program Listing

```
//UVAAE001 JOB 605B0,WANG ,MSGLEVEL×1


/*FORMAT PU,DDNAME×SYSPUNCH,DEST×LOCAL
/*MAIN  TIME×10,LINES×10
//    EXEC   LP
//LP.SYSIN  DD  *
          PROGRAM [!PM!]
          TITLE [! UVA EXAMPLES !]
          INITIALZ
          MOVE [XDATA,!EXMP1!]
          MOVE [XPBNAME,!LTMDYN!]
          CONVERT [!CHECK!,!SUMMARY!]
          BCDOUT [!ONE!]
          SETUP [!MIN!,!BOUND!,!BDS1!]
          PICTURE
          MOVE [XOBJ,!OBJF!]
          MOVE [XRHS,!RHS1!]
          PRIMAL
          SOLUTION
          MOVE [XDATA,!ANSWER!]
          PUNCH [!LIST!,!VALUE!]
          EXIT
          PEND
/*
//GO.SYSPRINT  DD  SYSOUT×A
//GO.SYSIN  DD  *
```

(Place MPS/360 Linear Programming Problem Data Deck here)

```
/*
```

The next two move instructions set data cells XDATA and XPBNAME. Identifiers beginning with X are predefined special variables for passing information to the procedures. The contents of XDATA, which is set to EXMP1, identifies the name of the data to be used by the following procedures. If the input stream has several named decks one can be picked for use by setting XDATA to the name on the deck NAME card. XPBNAME is the name to be given to the problem file — an internal MPS file, omitted from the system flowchart and designated PROBFILE.

Procedure CONVERT writes on PROBFILE in converted format the data deck named by XDATA. The contents of PBNAME identifies the converted data. Parameters 'CHECK' and 'SUMMARY' specify special checks to be performed and a summary of the data to be printed.

The next procedure, BCDOUT, prints the problem on the PROB file identified by XPBNAME. Parameter 'ONE' specifies one value per printed line.

Certain computations that must be done before the solution is found are done by SETUP; XPBNAME identifies the problem to be set up. Parameter 'MIN' specifies object function minimization; 'BOUND' and 'BDS' identifies BDSI as the vector name of the bounds specifications.

PICTURE causes a picture of the matrix to be printed.

The next two move statements identify vector OBJF to be used as the objective function coefficients and RHS1 the right-hand side vector.

PRIMAL is the optimization procedure. The next procedure SOLUTION produces a printout of the solution obtained by procedure PRIMAL.

The linear programming problem solution card file is produced by procedure PUNCH. Parameter 'LIST' produces a printed listing of the output and 'VALUE' is necessary for values to be punched.

Procedure EXIT is the final statement and returns control to the system from the program. It is equivalent to the FORTRAN CALL EXIT or STOP statements.

PEND, like FORTRAN END, specifies to the compiler the end of the program.

C. PSTPROC

A discussion of the program PSTPROC is given in this section. Appendix IV B contains a listing of program PSTPROC. There are three subroutines in PSTPROC in addition to the main section. Each of these is described below.

1. Main Section

The main section is described in the Section III of this report. Arrays have the same identifiers in both this program and PREPROC.

2. Subroutines

a. SEARCH

Subroutine SEARCH sets a parameter to 1 if a nonzero element is found in a given row of the passed array. If all row elements are zero, a 0 is returned.

b. TRAJ

Subroutine TRAJ plots the tabulation function against time; a tabulated listing is printed in addition to the line printer plot.

The logic of the program is straightforward; the position of the star is determined and computed GO TO statements transfer control to a WRITE-FORMAT pair which prints the star.

c. TRADE

The logic of this subroutine is much like that of TRAJ. It plots the tradeoff between the objective function values and the values of varying bounds.

APPENDIX IV

LISTINGS OF PROGRAMS

A. Program PREPROC

```
        PROGRAM   PREPROC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,PUNCH)
000003        DIMENSION IXX(8)
000003        COMMON /PLT/ ITR(8),ITD,KSWT
000003        COMMON /RWID/ ISP   ,MSP(8,2)
000003        COMMON/INIT/ S(1,20)
000003        COMMON /MXAP/ AEI(20,20),P(20,20)
000003        COMMON/ABD/A(20,20),B(20,4),D(20,4)
000003        COMMON/OBJ/PX1(5,20),PX2(5,4),PX3(5,4)
000003        COMMON/EKC/EM(10,10),CC(10,10),FF(10,4),CAY(10,10),UU(10,4)
000003        COMMON/IIST/ISET(10)
000003        COMMON/CONST/Y1(5,20),Y2(5,4),Y3(5,4)
000003        COMMON/TAB/Q1( 3,20),Q2( 3,3),Q3( 3,3)
000003        COMMON/UPLO/YU(20,5),YL(20,5),Y(20)
000003        COMMON/FOFUN/F(1,20,4)
000003        COMMON /TIT/ITITL(12)
000003        COMMON/ALL/ INITR,KS(8)
000003        COMMON /RWS/ NOBF,NOC,NOB,NOT
000003        COMMON /T/ TM
000003        COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE,IZ
000003        COMMON /MT/NOE
000003        COMMON/FORC/ NSETS
000003        COMMON /VER/IVE1,IVE2,IVE3,IVE4,IVE5,IVE6,IVE7,IVE8,IVE9,IVE10
000003        COMMON/VER1/IVE11,IVE12,IVE13,IVE14,IVE15,IVE16,IVE17,IVE19
000003        COMMON/VER2/IVE20,IVE21,IVE23,IVE24
000003        INM1  =(+6HA MATR)                                    1 24
000005        INM2  =(+6HB MATR)                                    1 25
000006        INM3  =(+6HD MATR)                                    1 26
000010        INM4  =(+6HM MATR)                                    1 27
000011        INM5  =(+6HC MATR)                                    1 28
000013        INM6  =(+6HK MATR)                                    1 29
000014        INM7  =(+6HF MATR)                                    1 30
000016        INM8  =(+6HU MATR)                                    1 31
000017        INM9  =(+6HINITIA)                                    1 32
000021        INM10 =(+6HFORCIN)                                    1 33
000022        INM11 = (+6HPX1 MA)                                   1 34
000024        INM12 = (+6HPX2 MA)                                   1 35
000025        INM13 = (+6HPX3 MA)                                   1 36
000027        INM14 = (+6HY1 MAT)                                   1 37
000030        INM15 = (+6HY2 MAT)                                   1 38
000032        INM16 = (+6HY3 MAT)                                   1 39
000033        INM17 = (+6HBOUNDS)                                   1 40
000035        INM18 = (+6HMODIFY)                                   1 41
000036        INM19 = (+6HQ1 MAT)                                   1 42
000040        INM20 = (+6HQ2 MAT)                                   1 43
000041        INM21 = (+6HQ3 MAT)                                   1 44
000043        INM22 = (+5HTRADE)                                    1 45
000044        INM23 = (+6HFINISH)                                   1 46
000046        INM24 = (+4HSTOP)                                     1 47
000047        INM25 = (+1H*)                                        1 48
000051        INM26 = (+6HRESTAR)                                   1 49
000052        INM27 = (+6HVERIFY)                                   1 50
000054        INM28 = (+6HTRAJEC)                                   1 51
000055      5 READ (5,10) (NPB,(ITITL (I) ,I = 1,12))
000071     10 FORMAT (1X,I5,4X,12A5)
000071        READ (5,20) NOE,MODE,NDF,NU,NF,NSETS,II,TM,NOB,NOC,ISP,NOT
000125     20 FORMAT (5X,I1,4X,I1,4X,5I5, F10.6,2I5,4X,I1,I5)
000125        WRITE (6,22) (NPB,(ITITL(I),I = 1,12))
```

```
000141      22 FORMAT (1H1,4X,16HPROBLEM NUMBER =,I5,5X,12A5)
000141         IF(NOE-1)23,23,28
000144      23 N=NDF
000146         M=N
000147         GO TO 26
000147      28 N=NDF*2
000151         M=N
000152      26 DO 2112 I=1,N
000154         S(1,I) = 0.0
000156         Y(I) = 0.0
000157         DO 2111 J=1,N
000160         A(I,J) = 0.0
000163         AEI(I,J)=0.0
000166           P(I,J)=0.0
000171    2111 CONTINUF
000173         DO 2119 J=1,4
000175         B(I,J) = 0.0
000200         D(I,J) = 0.0
000203    2119 CONTINUE
000205         DO 2112 J=1,5
000206         YU(I,J)=0.0
000211         YL(I,J)=0.0
000214    2112 CONTINUE
000220         DO 2116 I=1,NOC
000222         ISET(I) = 0
000223    2116 CONTINUE
000225         DO 2117 I=1,8
000227         IXX(I) = 0
000230         ITR(I) = 0
000231         MSP(I,1) = 0
000232         MSP(I,2) = 0
000233    2117 CONTINUE
000235         DO 2113 I=1,NDF
000236         DO 2113 J=1,NDF
000237         EM(I,J) = 0.0
000242         CC(I,J) = 0.0
000245         CAY(I,J) = 0.0
000247         FF(I,J) = 0.0
000252         UU(I,J) = 0.0
000254    2113 CONTINUE
000260         DO 2114 I=1,3
000262         DO 2114 J=1,4
000263         Q3(I,J) = 0.0
000266         Q2(I,J) = 0.0
000270         Y3(I,J) = 0.0
000272         Y2(I,J) = 0.0
000274         PX3(I,J) = 0.0
000276         PX2(I,J) = 0.0
000300    2114 CONTINUE
000303         DO 2115 I=1,3
000305         DO 2115 J=1,20
000306         Q1(I,J) = 0.0
000311         Y1 (I,J) = 0.0
000313         PX1(I,J) = 0.0
000315         F(1,J,I)=0.0
000320    2115 CONTINUE
000324          INITR=0
000325         IVE1 = 0
```

```
000326            IVE2  = 0
000327            IVE3  = 0
000330            IVE4  = 0
000331            IVE5  = 0
000332            IVE6  = 0
000333            IVE7  = 0
000334            IVE8  = 0
000335            IVE9  = 0
000336            IVE10 = 0
000337            IVE11 = 0
000340            IVE12 = 0
000341            IVE13 = 0
000342            IVE14 = 0
000343            IVE15 = 0
000344            IVE16 = 0
000345            IVE17 = 0
000346            IVE19 = 0
000347            IVE20 = 0
000350            IVE21 = 0
000351            IVE22 = 0
000352            IVE23 = 0
000353            IVE24 = 0
000354            NM = 0
000355            IZ = 0
000356            ITD = 0
000357            IVE24 = 1
000360            READ (5,24) (ISET(I),I = 1,NOT)
000372         24 FORMAT (6I10)
000372         25 READ (5,30 )NAME
000400         30 FORMAT (A6)
000400            IF (NAME .EQ. INM1 ) GO TO 445                          1124
000402            IF (NAME .EQ. INM2 ) GO TO 450                          1125
000404            IF (NAME .EQ. INM3 ) GO TO 455                          1126
000406            IF (NAME .EQ. INM4 ) GO TO 460                          1127
000410            IF (NAME .EQ. INM5 ) GO TO 465                          1128
000412            IF (NAME .EQ. INM6 ) GO TO 470                          1129
000414            IF (NAME .EQ. INM7 ) GO TO 475                          1130
000416            IF (NAME .EQ. INM8 ) GO TO 480                          1131
000420            IF (NAME .EQ. INM9 ) GO TO 45                           1132
000422            IF (NAME .EQ. INM10) GO TO 60                           1133
000424            IF (NAME .EQ. INM11) GO TO 125                          1134
000426            IF (NAME .EQ. INM12) GO TO 135                          1135
000430            IF (NAME .EQ. INM13) GO TO 140                          1136
000432            IF (NAME .EQ. INM14) GO TO 150                          1137
000434            IF (NAME .EQ. INM15) GO TO 160                          1138
000436            IF (NAME .EQ. INM16) GO TO 170                          1139
000440            IF (NAME .EQ. INM17) GO TO 200                          1140
000442            IF (NAME .EQ. INM18) GO TO 490                          1141
000444            IF (NAME .EQ. INM19) GO TO 500                          1142
000446            IF (NAME .EQ. INM20) GO TO 505                          1143
000450            IF (NAME .EQ. INM21) GO TO 510                          1144
000452            IF (NAME .EQ. INM22) GO TO 515                          1145
000454            IF (NAME .EQ. INM23) GO TO 860                          1146
000456            IF (NAME .EQ. INM24) GO TO 999                          1147
000460            IF (NAME .EQ. INM25) GO TO 25                           1148
000462            IF (NAME .EQ. INM26) GO TO 5                            1149
000464            IF (NAME .EQ. INM27) GO TO 550                          1150
000466            IF (NAME .EQ. INM28) GO TO 580                          1151
```

```
000470      40 WRITE (6,32) NAME
000476      32 FORMAT (25H0 DO NOT RECOGNIZE NAME ≠,A6,1H≠)
000476         GO TO 999
           C    INITIAL CONDITIONS
000477      45 READ (5,50) (KEY,I,S(1,I))
000512      50 FORMAT (I1,I5,5X,F12.6)
000512         IVE9 = 1
000513         IF (KEY) 25,45,25
           C   FORCING FUNCTIONS
000515      60 READ (5,65) KEY2,I
000525      65 FORMAT (I1,I5)
000525         IVE10= 1
000526      70 READ (5,75) KEY1,J,KIND
000540      75 FORMAT (I1,I5,4X,I1)
000540         IF (KIND .EQ.1) GO TO 83
000542      80 READ (5,50)(KEY,K,F(I,K,J))
000557         IF (KEY) 105,80,105
000561      83 WRITE (6,85)
000565      85 FORMAT (22H READ FUNCTION OF TIME)
000565     105 IF (KEY1 .EQ.0) GO TO 70
000566         IF (KEY2) 25,60,25
           C   OBJECTIVE FUNCTION
000570     125 READ (5,130) KEY,I,J,PX1(I,J)
000606     130 FORMAT (I1,2I5,F12.6)
000606         IVE11= 1
000607         IF (KEY) 25,125,25
000611     135 READ (5,130) KEY,I,J,PX2(I,J)
000627         IVE12= 1
000630         IF (KEY) 25,135,25
000632     140 READ (5,130) KEY,I,J,PX3(I,J)
000650         IVE13= 1
000651         IF (KEY) 25,140,25
           C   CONSTRAINT
000653     150 READ (5,130) KEY,I,J,Y1(I,J)
000671         IVE14= 1
000672         IF (KEY) 25,150,25
000674     160 READ (5,130) KEY,I,J,Y2(I,J)
000712         IVE15= 1
000713         IF (KEY) 25,160,25
000715     170 READ (5,130) KEY,I,J,Y3(I,J)
000733         IVE16= 1
000734         IF (KEY) 25,170,25
           C   BOUNDS OF CONSTRAINTS
000736     185 FORMAT (F12.6,I5,F12.6)
000736     200 READ (5,205) KEY1,I,MSP(I,1),MSP(I,2)
000752     205 FORMAT (I1,I5,4X,I1,4X,I1)
000752         IVE17= 1
000753         IF (MSP(I,1)-1) 210,290,350
000756     210 IF( MSP(I,2) -1)215,235,250
000761     215 READ (5,218 )NCN
000767     218 FORMAT (1X,I1)
000767         IF (NCN-1) 220,224,224
000772     220 READ (5,185) SVA1,NIC1,VIC1
001004         DO 222 K = 1,II
001006     222 YU(K,I) = SVA1
001015         IF (NIC1) 225,225,223
001016     223 IXX(I) = 1
001020         IZ = NIC1
```

```
001022        SZ1 = SVA1
001023        VZ1 = VIC1
001025        GO TO 225
001025    224 READ( 5,50) KEY,K,YU(K,I)
001041        IF (KEY .EQ. 0) GO TO 224
001042    225 READ (5,218) NCN
001050        IF (NCN-1) 226,230,230
001053    226 READ (5,185) SVA2,NIC2,VIC2
001065        DO 227 K = 1,II
001067    227 YL(K,I) = SVA2
001076        IF (NIC2) 440,440,229
001077    229 IXX(I) = IXX(I)+2
001102        SZ2 = SVA2
001103        VZ2 = VIC2
001104        IZ = NIC2
001106        GO TO 440
001106    230 READ (5,50) KEY,K,YL(K,I)
001122        IF( KEY) 440,230,440
001124    235 READ(5,218) NCN
001132        IF (NCN-1) 240,245,245
001135    240 READ(5,185) SVA1,NIC1,VIC1
001147        DO 242 K = 1,II
001151    242 YU(K,I) = SVA1
001160        IF (NIC1) 440,440,243
001161    243 IXX(I) = 1
001163        IZ = NIC1
001165        SZ1 = SVA1
001166        VZ1 = VIC1
001170        GO TO 440
001170    245 READ(5,50) KEY,K,YU(K,I)
001204        IF (KEY) 440,245,440
001206    250 IF (MSP(I,2) -2) 225,225,235
001212    290 IF (MSP(I,2)-1)292,305,315
001215    292 READ(5,218) NCN
001223        IF (NCN-1) 295,300,300
001226    295 READ (5,293) SVA1,NIC1,VIC1,KS(I)
001242    293 FORMAT (F12.6,I5,F12.6,I5)
001242        K = KS(I)
001244        YU(K,I) = SVA1
001250        IF (NIC1) 301,301,299
001251    299 IXX(I) = 1
001253        IZ = NIC1
001255        SZ1 = SVA1
001256        VZ1 = VIC1
001260        GO TO 301
001260    300 READ (5,50) KEY,K,YU(K,I)
001274        KS (I) = K
001276        IF (KEY .EQ. 0) GO TO 300
001277    301 READ (5,218) NCN
001305        IF (NCN-1) 302,304,304
001310    302 READ(5,293) SVA2,NIC2,VIC2,KS(I)
001324        K = KS(I)
001326        YL(K,I) = SVA2
001332        IF (NIC2) 440,440,303
001333    303 IXX(I) = IXX(I)+2
001336        SZ2 = SVA2
001337        VZ2 = VIC2
001340        IZ = NIC2
```

```
001342          GO TO 440
001342      304 READ (5,50) KEY,K,YL(K,I)
001356          KS(I) = K
001360          IF (KEY) 440,304,440
001362      305 READ (5,218) NCN
001370          IF (NCN-1) 307,310,310
001373      307 READ (5,293) SVA1,NIC1,VIC1,KS(I)
001407          K = KS(I)
001411          YU(K,I) = SVA1
001415          IF( NIC1) 440,440,308
001416      308 IXX(I) = 1
001420          IZ = NIC1
001422          SZ1 = SVA1
001423          VZ1 = VIC1
001425          GO TO 440
001425      310 READ (5,50) KEY,K,YU(K,I)
001441          KS(I) = K
001443          IF (KEY) 440,310,440
001445      315 IF (MSP(I,2)-2) 301,301,305
001451      350 IF (MSP(I,2)-2) 360,370,380
001454      360 CONTINUE
001454      370 CONTINUE
001454      380 CONTINUE
001454          READ (5,390) KI,KJ
001464      390 FORMAT (2I5)
001464      440 IF (KEY1) 25,200,25
          C     A MATRIX
001466      445 READ(5,130)  KEY,I,J,A(I,J)
001504          IVE1 = 1
001505          IF (KEY) 25,445,25
          C     B MATRIX
001507      450 READ (5,130)KEY,I,J,B(I,J)
001525          IVE2 = 1
001526          IF (KEY) 25,450,25
          C     D MATRIX
001530      455 READ (5,130) KEY,I,J,D(I,J)
001546          IVE3 = 1
001547          IF (KEY) 25,455,25
          C     M MATRIX
001551      460 READ (5,130) KEY,I,J,EM(I,J)
001567          IVE4 = 1
001570          IF (KEY) 25,460,25
          C     C MATRIX
001572      465 READ (5,130) KEY,I,J,CC(I,J)
001610          IVE5 = 1
001611          IF (KEY) 25,465,25
          C     K MATRIX
001613      470 READ (5,130) KEY,I,J,CAY(I,J)
001631          IVE6 = 1
001632          IF (KEY) 25,470,25
          C     F MATRIX
001634      475 READ (5,130) KEY,I,J,FF(I,J)
001652          IVE7 = 1
001653          IF (KEY) 25,475,25
          C     U MATRIX
001655      480 READ (5,130) KEY,I,J,UU(I,J)
001673          IVE8 = 1
001674          IF (KEY) 25,480,25
```

```
              C     MODIFY PROBLEM
 001676      490 NM = NM+1
 001700          WRITE (6,495) NM,(ITITL(I),I = 1,12)
 001713      495 FORMAT (1H1,10X,16HMODIFICATION NO.,I5,10X,12A5)
 001713          INT = 0
 001714          INP = 0
 001715          IVE23 = 0
 001716          GO TO 25
              C    TABULATING
 001716      500 READ (5,130 )KEY,I,J,Q1(I,J)
 001734          IVE19= 1
 001735          IF (KEY) 25,500,25
 001737      505 READ (5,130 )KEY,I,J,Q2(I,J)
 001755          IVE20= 1
 001756          IF (KEY) 25,505,25
 001760      510 READ (5,130 )KEY,I,J,Q3(I,J)
 001776          IVE21= 1
 001777          IF (KEY) 25,510,25
              C    TRADE OFF DIAGRAM
 002001      515 READ (5,555) ITD,KSHT
 002011          GO TO 25
 002012      550 READ (5,555) INP,INT
 002022      555 FORMAT (2I5)
 002022          IF (INP .EQ. 0 ) GO TO 560
 002023          CALL VERFY
 002024      560 IF (INT .EQ. 0) GO TO 25
 002025          IF (IVE23 .EQ.0 ) GO TO 25
 002026          CALL VERFY2
 002027          GO TO 25
 002030      580 READ (5,75) KEY,I,ITR(I)
 002042          IF (KEY) 25,580,25
 002044      860 IF (NOE .EQ.-1) GO TO 865
 002046          CALL COVERT
 002047      865 IVE 23=1
 002050          CALL MEXAP
 002051          CALL BRIDGE
 002052          CALL ELEMTP
 002053          CALL ELEMTQ
 002054          CALL RHDATA
 002055          CALL POST
              C    CALL LPDATA
 002056          CALL PRNT
 002057          IF (INT .EQ. 0) GO TO 870
 002060          CALL VERFY2
 002061      870 IF (IZ .EQ. 0) GO TO 25
 002062      880 I = 1
 002063          IZZ = IZ+1
 002065      890 IF (IXX(I)) 910,1140,910
 002067      910 IR = I
 002071          IF (IXX(I)-2) 920,940,960
 002073      920 DO 922 L = 1,IZZ
 002075      922 Y(L) = SZ1+(L-1)*VZ1
 002105          IF (MSP(I,1) .EQ. 0) GO TO 930
 002107          ID = 2
 002110          GO TO 1110
 002110      930 ID = 1
 002111          GO TO 980
 002112      940 DO 942 L = 1,IZZ
```

```
002114       942 Y(L) = SZ2+(L-1)*VZ2
002124           IF (MSP(I,1) .EQ. 0 ) GO TO 950
002126           ID = 4
002127           GO TO 1110
002127       950 ID = 3
002130           GO TO 980
002131       960 DO 962 L = 1,IZZ
002133       962 Y(L) = SZ1+(L-1)*VZ1
002143           IF (MSP(I,1) .EQ. 0) GO TO 970
002145           ID = 6
002146           GO TO 1110
002146       970 ID = 5
002147       980 DO 1100 L = 1,IZ
002151           II1=II
002153           IF (ID -3) 982,986,990
002155       982 DO 984 K = 1,II1
002157           YU(K,IR) = SZ1+L*VZ1
002166       984 CONTINUE
002170           GO TO 994
002171       986 DO 988 K = 1,II1
002173           YL(K,IR) = SZ2+L*VZ2
002202       988 CONTINUE
002204           GO TO 994
002205       990 DO 992 K = 1,II1
002207           YU(K,IR) = SZ1+L*VZ1
002215           YL(K,IR) = SZ2+L*VZ2
002222       992 CONTINUE
002224       994 CALL ELEMTP
002225           CALL ELEMTQ
002226           PUNCH 789
002232       789 FORMAT(3HRHS)
002232           CALL RHDATA
002233           CALL PRNT
002234           IF (INP .EQ. 0) GO TO 1100
002235           CALL VERFY
002236      1100 CONTINUE
002241           GO TO 1140
002241      1110 DO 1130 L = 1,IZ
002243           IR2 = KS(IR)
002245           IF (ID -4) 1112,1114,1116
002250      1112 YU(IR2,IR) = SZ1+L*VZ1
002257           GO TO 1120
002257      1114 YL(IR2,IR) = SZ2+L*VZ2
002266           GO TO 1120
002266      1116 YU(IR2,IR) = SZ1+L*VZ1
002275           YL(IR2,IR) = SZ2+L*VZ2
002302      1120 CALL ELEMTP
002303           CALL ELEMTQ
002304           PUNCH 789
002310           CALL RHDATA
002311           CALL PRNT
002312           IF (INP .EQ. 0) GO TO 1130
002313           CALL VERFY
002314      1130 CONTINUE
002317           GO TO 25
002317      1140 IF (I-NOC) 1150,25,25
002322      1150 I = I+1
002324           GO TO 890
```

```
                SUBROUTINE POST
000002          COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE,IZ
000002          COMMON /RWS/ NOBF,NOC,NOB,NOT
000002          COMMON/FORC/ NSETS
000002          COMMON /PLT/ ITR(8),ITD,KSWT
000002          COMMON/IIST/ISET(10)
000002          COMMON/TAB/Q1( 3,20),Q2( 3,3),Q3( 3,3)
000002          COMMON/BRG/R(20,20,4),T(20,20,4)
000002          COMMON/CSS/C1(20,20),C2(1,20,20),C(1,20,20)
000002          COMMON/FOFUN/F(1,20,3)
000002          COMMON/T/TM
000002          COMMON/UPLO/YU(20,5),YL(20,5),Y(20)
000002          II1 = II-1
000004          IZZ = IZ+1
000006          PUNCH 10,II,N,NF,NU,NDF,NSETS,NOT,IZZ,ITD,KSWT,TM
000037       10 FORMAT(10I5,F10.6)
000037       30 FORMAT (6E12.5)
000037          DO 20 I = 1,NOT
000041       20 PUNCH 30,(Q1(I,J),J = 1,N)
000057          DO  40 I = 1,NOT
000060       40 PUNCH 30,(Q2(I,J),J = 1,NU)
000076          DO 50 I = 1,NOT
000077       50 PUNCH 30,(Q3(I,J) ,J = 1,NF)
000115          PUNCH 30,(Y(I),I = 1,IZZ)
000127          DO 80 K = 1,II1
000131          DO 80 I = 1,N
000132       80 PUNCH 30,(R(K,I,J),J = 1,NU)
000155          DO 90 IS = 1,NSETS
000157          DO 90 K = 1,II1
000160       90 PUNCH 30,(F(IS,K,I) ,I = 1,NF)
000201          DO 100 IS = 1,NSETS
000203          DO 100 K = 2,II
000204      100 PUNCH 30, (C(IS,K,I) ,I = 1,N)
000225          PUNCH 10,(ISET(I), I = 1,NOT)
000240          PUNCH 10,(ITR(I),I = 1,NOT)
000253          RETURN
000254          END
```

```
        SUBROUTINE VERFY
000002        COMMON /RWID/ ISP    ,MSP(8,2)
000002        COMMON/INIT/ S(1,20)
000002        COMMON/EKC/EM(10,10),CC(10,10),FF(10,4),CAY(10,10),UU(10,4)
000002        COMMON/UPLO/YU(20,5),YL(20,5),Y(20)
000002        COMMON/FOFUN/F(1,20,3)
000002        COMMON/ABD/A(20,20),B(20,4),D(20,4)
000002        COMMON/OBJ/PX1(5,20),PX2(5,4),PX3(5,4)
000002        COMMON/CONST/Y1(5,20),Y2(5,4),Y3(5,4)
000002        COMMON/IIST/ISET(10)
000002        COMMON/TAB/Q1( 3,20),Q2( 3,3),Q3( 3,3)
000002        COMMON /RWS/ NOBF,NOC,NOB,NOT
000002        COMMON /INPT/II,NZ,M,NF,NU,KI,KJ,NOF,MODE
000002        COMMON/FORC/ NSETS
000002        COMMON /VER/IVE1,IVE2,IVE3,IVE4,IVE5,IVE6,IVE7,IVE8,IVE9,IVE10
000002        COMMON/VER1/IVE11,IVE12,IVE13,IVE14,IVE15,IVE16,IVE17,IVE19
000002        COMMON/VER2/IVE20,IVE21,IVE23,IVE24
000002        N = NOF
000004        WRITE (6,5)
000007      5 FORMAT (27H1VERIFICATION OF INPUT DATA)
000007        IF (IVE1 .EQ. 0) GO TO 40
000010        WRITE (6,10)
000014     10 FORMAT (9H0A MATRIX)
000014        DO 20 I = 1,M
000016     20 WRITE (6,30) (A(I,J),J = 1,M)
000035     30 FORMAT (1X,10E12.5,(/6X,9E12.5))
000035     25 FORMAT(5X,2I5)
000035     35 FORMAT (1X,2HK=,I4,10E12.5,(/ 6X,9E12.5))
000035     40 IF (IVE2 .EQ. 0) GO TO 70
000036        WRITE (6,50)
000042     50 FORMAT (9H0B MATRIX)
000042        DO 60 I = 1,M
000044     60 WRITE (6,30) (B(I,J),J = 1,NU)
000063     70 IF (IVE3 .EQ. 0) GO TO 90
000064        WRITE (6,75)
000070     75 FORMAT (9H0D MATRIX)
000070        DO 80 I = 1,M
000072     80 WRITE (6,30) (D(I,J),J = 1,NF)
000111     90 IF (IVE4 .EQ. 0) GO TO 110
000112        WRITE (6,95)
000116     95 FORMAT (9HCM MATRIX)
000116        DO 100 I = 1,N
000120    100 WRITE (6,30) (EM(I,J),J = 1,N)
000136    110 IF (IVE5 .EQ. 0) GO TO 130
000137        WRITE (6,115)
000143    115 FORMAT (9H0C MATRIX)
000143        DO 120 I = 1,N
000145    120 WRITE (6,30) (CC(I,J),J = 1,N )
000163    130 IF (IVE6 .EQ. 0) GO TO 150
000164        WRITE (6,135)
000170    135 FORMAT (9HCK MATRIX)
000170        DO 140 I = 1,N
000172    140 WRITE (6,30)(CAY(I,J),J = 1,N )
000210    150 IF (IVE7 .EQ. 0) GO TO 170
000211        WRITE (6,155)
000215    155 FORMAT (9H0F MATRIX)
000215        DO 160 I = 1,N
```

```
000217    160 WRITE (6,30) (FF(I,J),J = 1,NF)
000235    170 IF (IVE8 .EQ. 0) GO TO 190
000236        WRITE (6,175)
000242    175 FORMAT (9H0U MATRIX)
000242        DO 180 I = 1,N
000244    180 WRITE (6,30) (UU(I,J),J = 1,NU)
000262    190 IF (IVE9 .EQ. 0) GO TO 200
000263        WRITE (6,195)
000267    195 FORMAT (19H0INITIAL CONDITIONS)
000267        WRITE (6,30) (S(1,I) ,I = 1,M)
000302    200 IF (IVE10.EQ. 0) GO TO 225
000303        WRITE (6,205)
000307    205 FORMAT (17H0FORCING FUNCTION)
000307        DO 220 IS = 1,NSETS
000311        WRITE (6,210)
000314    210 FORMAT (1H0)
000314        II1 = II-1
000316        DO 220 K = 1,II1
000320    220 WRITE (6,30) (F(IS,K,J),J = 1,NF)
000341    225 IF (IVE11.EQ. 0) GO TO 240
000342        WRITE (6,230)
000346    230 FORMAT (11H0PX1 MATRIX)
000346        DO 235 I = 1,NOB
000350    235 WRITE (6,30) (PX1(I,J),J = 1,M)
000366    240 IF (IVE12.EQ. 0) GO TO 260
000367        WRITE (6,245)
000373    245 FORMAT (11H0PX2 MATRIX)
000373        DO 250 I = 1,NOB
000375    250 WRITE (6,30) (PX2(I,J),J = 1,NU)
000413    260 IF (IVE13.EQ. 0) GO TO 280
000414        WRITE (6,265)
000420    265 FORMAT (11H0PX3 MATRIX)
000420        DO 270 I = 1,NOB
000422    270 WRITE (6,30) (PX3(I,J),J = 1,NF)
000440    280 IF (IVE14.EQ. 0) GO TO 300
000441        WRITE (6,285)
000445    285 FORMAT (10H0Y1 MATRIX)
000445        DO 290 I = 1,NOC
000447    290 WRITE (6,30) ( Y1(I,J),J = 1,M)
000465    300 IF (IVE15.EQ. 0) GO TO 320
000466        WRITE (6,305)
000472    305 FORMAT (10H0Y2 MATRIX)
000472        DO 310 I = 1,NOC
000474    310 WRITE (6,30) ( Y2(I,J),J = 1,NU)
000512    320 IF (IVE16.EQ. 0) GO TO 340
000513        WRITE (6,325)
000517    325 FORMAT (10H0Y3 MATRIX)
000517        DO 330 I = 1,NOC
000521    330 WRITE (6,30) ( Y3(I,J),J = 1,NF)
000537    340 IF (IVE17.EQ. 0) GO TO 380
000540        WRITE (6,345)
000544    345 FORMAT (13H0UPPER BOUNDS)
000544        DO 350 K = 1,II
000546    350 WRITE (6,35)  K,(YU(K,I),I = 1,NOC)
000567        WRITE (6,355)
000572    355 FORMAT (13H0LOWER BOUNDS)
000572        DO 360 K = 1,II
000574    360 WRITE (6,35)  K,(YL(K,I),I = 1,NOC)
```

```
000615         WRITE (6,365)
000620     365 FORMAT(21H0SPECIFICATION MATRIX)
000620         DO 370 I = 1,NOC
000622     370 WRITE( 6,25) (MSP(I,J),J = 1,2)
000637     380 IF (IVE19.EQ. 0) GO TO 400
000640         WRITE (6,385)
000644     385 FORMAT (10H0Q1 MATRIX)
000644         DO 390 I = 1,NOT
000646     390 WRITE (6,30) (Q1(I,J),J = 1,M)
000664     400 IF (IVE20.EQ. 0) GO TO 420
000665         WRITE (6,405)
000671     405 FORMAT (10H0Q2 MATRIX)
000671         DO 410 I = 1,NOT
000673     410 WRITE (6,30) (Q2(I,J),J = 1,NU)
000711     420 IF (IVE21.EQ. 0) GO TO 460
000712         WRITE (6,425)
000716     425 FORMAT (10H0Q3 MATRIX)
000716         DO 430 I = 1,NOT
000720     430 WRITE (6,30) (Q3(I,J),J = 1,NF)
000736     460 IF (IVE24 .EQ.0) GO TO 470
000737         WRITE (6,465)
000743     465 FORMAT (12H0ISET VECTOR)
000743         WRITE (6,25) (ISET (I),I = 1,NOT)
000756     470 RETURN
000757         END
```

```
                SUBROUTINE VERFY2
000002          COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE,IZ
000002          COMMON /RWS/ NOBF,NOC,NOB,NOT
000002          COMMON/FORC/ NSETS
000002          COMMON /MT/NOE
000002          COMMON/PRG/R(20,20,4),T(20,20,4)
000002          COMMON/CSS/C1(20,20),C2(1,20,20),C(1,20,20)
000002          COMMON/ABD/A(20,20),B(20,4),D(20,4)
000002          COMMON /MXAP/ AEI(20,20),P(20,20)
000002          WRITE (6,20)
000006       20 FORMAT (31H1DATA CALCULATED WITHIN PROGRAM)
000006       30 FORMAT (1X,10E12.5,(/11X,9E12.5))
000006          IF (NOE .EQ. 1) GO TO 35
000010          WRITE (6,200)
000014      200 FORMAT (9H0A MATRIX)
000014          DO 210 I = 1,M
000016      210 WRITE (6,30) (A(I,J),J = 1,M)
000035          WRITE (6,220)
000040      220 FORMAT (9H0B MATRIX)
000040          DO 230 I = 1,M
000042      230 WRITE (6,30) (B(I,J),J = 1,NU)
000061          WRITE (6,240)
000064      240 FORMAT (9H0D MATRIX)
000064          DO 250 I = 1,M
000066      250 WRITE (6,30) (D(I,J),J = 1,NF)
000105       35 WRITE (6,40)
000111       40 FORMAT (9H0MATRIX R)
000111          II1 = II-1
000113          DO 60 K = 1,II1
000115          WRITE (6,50)
000120       50 FORMAT (1H0)
000120          DO 60 I = 1,M
000122       60 WRITE (6,30) (R(K,I,J),J = 1,NU)
000145          WRITE (6,70)
000151       70 FORMAT (9H0MATRIX T)
000151          DO 80 K = 1,II1
000153          WRITE (6,50)
000156          DO 80 I = 1,M
000160       80 WRITE (6,30) (T(K,I,J),J = 1,NF)
000203          WRITE (6,90)
000207       90 FORMAT (9H0MATRIX C)
000207          DO 100 IS = 1,NSETS
000211          WRITE (6,50)
000214          DO 100 K = 2,II
000216      100 WRITE (6,30) (C(IS,K,I),I = 1,M)
000237          WRITE (6,110)
000243      110 FORMAT (9H0MATRIX P)
000243          DO 120 I = 1,M
000245      120 WRITE (6,30) (P(I,J),J = 1,M)
000264          WRITE (6,130)
000267      130 FORMAT (11H0MATRIX AEI)
000267          DO 140 I = 1,M
000271      140 WRITE (6,30) (AEI(I,J), J = 1,M)
000310          RETURN
000310          END
```

```
              SUBROUTINE COVERT
000002        COMMON/EKC/EM(10,10),CC(10,10),FF(10,4),CAY(10,10),UU(10,4)
000002        DIMENSION EMI(10,10)
000002         DIMENSION CM(10,10),CYM(10,10),UM(10,10),FM(10,10)
000002        COMMON/ABD/A(20,20),B(20,4),D(20,4)
000002        COMMON /INPT/II,NZ,M,NF,NU,KI,KJ,NDF,MODE
000002        IF(NDF-1)90,80,90
000004     80 EMI(1,1) = 1./EM(1,1)
000006        N=NDF
000010        GO TO 100
000010     90 CALL MIV(EM,NDF,EMI,10)
000013        N=NDF
000015    100 CALL M22X2 (N,N,N,EMI,CC,CM)
000021        CALL M22X2 (N,N,N,EMI,CAY,CYM)
000025        CALL M22X2(N,N,NU,EMI,UU,UM)
000031        CALL M22X2(N,N,NF,EMI,FF,FM)
000035        DO 125 I= 1,N
000037        A(I+N,I) = 1.0
000044        DO 125 J= 1,N
000045        A(I,J) =  CM(I,J)*(-1.0)
000053        A(I,J+N) = CYM(I,J)*(-1.0)
000061        IF (J .GT. NF) GO TO 120
000064        D(I,J  ) =  FM(I,J)
000071    120 IF (J .GT. NU) GO TO 125
000075        B(I,J  ) =  UM(I,J)*(-1.0)
000103    125 CONTINUE
000110        RETURN
000110        END
```

```
                SUBROUTINE BRIDGE
000002          COMMON/INIT/ S(1,20)
000002          DIMENSION CT(1,20,20)
000002          COMMON/BRG/R(20,20,4),T(20,20,4)
000002          COMMON/CSS/C1(20,20),C2(1,20,20),C(1,20,20)
000002          COMMON/FOFUN/F(1,20,3)
000002          COMMON/ABD/A(20,20),B(20,4),D(20,4)
000002          COMMON /MXAP/ AEI(20,20),P(20,20)
000002          COMMON/FORC/ NSETS
000002          COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE
000002          DO 220 I=1,N
000004          C1(1,I)=S(1,I)
000010          DO 210 J=1,NU
000012          R(1,I,J)=0.0
000017          DO 210 L=1,N
000020          R(1,I,J)=R(1,I,J)+AEI(I,L)*B(L,J)
000034      210 CONTINUE
000041          DO 220 J=1,NF
000042          T(1,I,J)=0.0
000047          DO 220 L=1,N
000050          T(1,I,J)=T(1,I,J)+AEI(I,L)*D(L,J)
000064      220 CONTINUE
000073          DO 500 K=2,II
000075          K4=K-1
000077          DO 330 I=1,N
000100          DO 310 J=1,NU
000101          R(K,I,J)=0.0
000106          DO 310 L=1,N
000110          R(K,I,J)=R(K,I,J)+P(I,L)*R(K4,L,J)
000127      310 CONTINUE
000134          DO 320 J=1,NF
000135          T(K,I,J)=0.0
000142          DO 320 L=1,N
000144          T(K,I,J)=T(K,I,J)+P(I,L)*T(K4,L,J)
000163      320 CONTINUE
000170          C1(K,I)=0.0
000173          DO 330 J=1,N
000175          C1(K,I)=C1(K,I)+P(I,J)*C1(K4,J)
000210      330 CONTINUE
000214          DO 430 IS=1,NSETS
000216          DO 390 L=1,N
000217          C2(IS,K,L)=0.0
000224      390 CONTINUE
000226          DO 400 I=1,K4
000230          KMI=K-I
000232          DO 410 IP=1,N
000233          CT(IS,K,IP)=0.0
000237          DO 410 J=1,NF
000241           CT(IS,K,IP)=CT(IS,K,IP)+T(KMI,IP,J)*F( IS,I,J)
000260      410 CONTINUE
000265          DO 420 L1=1,N
000266          C2(IS,K,L1)=C2(IS,K,L1)+CT(IS,K,L1)
000277      420 CONTINUE
000301      400 CONTINUE
000303          DO 430 I=1,N
000305          C(IS,K,I)=C1(K,I)+C2(IS,K,I)
000317      430 CONTINUE
```

```
000326        RETURN
000327        END
```

```
              SUBROUTINE PRNT
000002        COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NOF,MODE
000002        COMMON/PVAL/H(120,80)
000002        COMMON/RHV/G(120),NRE(120)
000002        COMMON/ALL/ INITR,KS(8)
000002    251 JL = (II-1)*NU+1
000006        WRITE (6,225)
000011    225 FORMAT (9H0MATRIX H/1X,4HROWS)
000011        DO 200 I = 1,INITR
000013    200 WRITE (6,250) (I,(H(I,J),J = 1,JL))
000034    250 FORMAT (1X,I4,5X, 7E14.7,(/15X,7E14.7))
000034        WRITE (6,350)
000037    350 FORMAT (26H1RHS VALUES AND EQUALITIES/1X,4HROWS)
000037        DO 400 I = 1,INITR
000041    400 WRITE(6,450) I,NRE(I),G(I)
000055    450 FORMAT (1X,I4,3X,A1,3X,E12.5)
000055        DO 500 I = 1,INITR
000056        G(I) = 0.0
000057        NRE(I) = 0
000060        DO 500 J = 1,JL
000062        H(I,J) = 0.0
000066    500 CONTINUE
000073        INITR = 0
000073        RETURN
000074        END
```

```
                SUBROUTINE LPDATA
000002          MPSNM = 5HEXMP1
000004          PUNCH 610,MPSNM
000011      610 FORMAT (4HNAME,10X,A6)
000011          PUNCH 612
000015      612 FORMAT (4HROWS)
000015          CALL RODATA
000016          PUNCH 620
000022      620 FORMAT (7HCOLUMNS)
000022          CALL CODATA
000023          PUNCH 625
000027      625 FORMAT (3HRHS)
000027          CALL RHDATA
000030          PUNCH 630
000034      630 FORMAT(6HBOUNDS)
000034          CALL BODATA
000035          PUNCH 660
000041      660 FORMAT(6HENDATA)
000041          RETURN
000042          END
```

```
                SUBROUTINE RODATA
000002          COMMON/RHV/G(120),NRE(120)
000002          COMMON/ALL/ INITR,KS(8)
000002          ILAST = INITR
000003          CHK = 3.
000005          IFRST = 9
000006          ISEC = 99
000007          IF (ILAST -9) 10,10,20
000012       10 IFRST = ILAST
000013          CHK = 1.
000015          GO TO 40
000016       20 IF(ILAST - 99) 30,30,40
000021       30 ISEC = ILAST
000022          CHK = 2.
000024       40 IFRS1 = IFRST+1
000026          ISEC2 = ISEC+1
000030          PUNCH 42
000033       42 FORMAT (1X,1HN,2X,4HOBJF)
000033          DO 50 I = 1,IFRST
000035       45 PUNCH 55,NRE(I),I
000045       50 CONTINUE
000050       55 FORMAT (1X,A1,2X,3HROW,I1)
000050          IF(CHK-1.) 100,100,60
000052       60 DO 65 I = IFRS1,ISEC
000054       63 PUNCH 70,NRE(I),I
000064       65 CONTINUE
000067       70 FORMAT(1X,A1,2X,3HROW,I2)
000067          IF(CHK -2.) 100,100,80
000071       80 DO 85 I = ISEC2,ILAST
000073       83 PUNCH 90,NRE(I),I
000103       85 CONTINUE
000106       90 FORMAT(1X,A1,2X,3HROW,I3)
000106      100 RETURN
000107          END
```

```
              SUBROUTINE CODATA
000002        COMMON/PVAL/H(120,80)
000002        COMMON/ALL/ INITR,KS(8)
000002        COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE
000002    253 FORMAT(4X,3HCOL,I3,4X,3HROW,I3,4X,E12.5)
000002     85 FORMAT (4X,4HCOL1,6X,4HOBJF,6X,E12.5)
000002     95 FORMAT(4X,3HCOL,I1,6X,3HROW,I1,6X,E12.5)
000002    115 FORMAT(4X,3HCOL,I1,6X,3HROW,I2,5X,E12.5)
000002    133 FORMAT (4X,3HCOL,I1,6X,3HROW,I3,4X,E12.5)
000002    163 FORMAT (4X,3HCOL,I2,5X,3HROW,I1,6X,E12.5)
000002    182 FORMAT(4X,3HCOL,I2,5X,3HROW,I2,5X,E12.5)
000002    223 FORMAT(4X,3HCOL,I3,4X,3HROW,I1,6X,E12.5)
000002    202 FORMAT (4X,3HCOL,I2,5X,3HROW,I3,4X,E12.5)
000002    238 FORMAT(4X,3HCOL,I3,4X,3HROW,I2,5X,E12.5)
000002        ILAST = INITR
000004        JLAST = (II-1)*NU+1
000010        CHJ = 3.
000011        CHK = 3.
000012        IFRST = 9
000013        ISEC = 99
000014        IF (ILAST -9) 10,10,20
000016     10 IFRST = ILAST
000020        CHK = 1.
000021        GO TO 40
000022     20 IF(ILAST -99) 30,30,40
000025     30 ISEC = ILAST
000027        CHK = 2.
000030     40 JFRST = 9
000031        JSEC = 99
000032        IF(JLAST-9) 50,50,60
000035     50 JFRST = JLAST
000037        CHJ = 1.
000040        GO TO 80
000041     60 IF(JLAST-99) 70,70,80
000044     70 JSEC = JLAST
000046        CHJ = 2.
000047     80 IFRS1 = IFRST+1
000051        ISEC2 = ISEC+1
000053        JFRS1 = JFRST+1
000054        JSEC2 = JSEC+1
000056        XOBJ = 1.0
000057        PUNCH 85,XOBJ
000065        DO 130 J = 1,JFRST
000067        DO 93 I = 1,IFRST
000070        IF( H(I,J))90,93,90
000074     90 PUNCH 95,J,I, H(I,J)
000110     93 CONTINUE
000113        IF(CHK-1.) 130,130,100
000115    100 DO 113 I = IFRS1,ISEC
000117        IF( H(I,J))110,113,110
000123    110 PUNCH 115,J,I, H(I,J)
000137    113 CONTINUE
000142        IF(CHK-2.) 130,130,120
000144    120 DO 128 I = ISEC2,ILAST
000146        IF( H(I,J))125,128,125
000152    125 PUNCH 133,J,I, H(I,J)
000166    128 CONTINUE
```

```
000171      130 CONTINUE
000174      140 IF(CHJ-1.) 260,260,150
000177      150 DO 200 J = JFRS1,JSEC
000201          DO 160 I = 1,IFRST
000202          IF( H(I,J))158,160,158
000206      158 PUNCH 163,J,I, H(I,J)
000222      160 CONTINUE
000225          IF(CHK -1.) 200,200,170
000227      170 DO 180 I = IFRS1,ISEC
000231          IF( H(I,J))175,180,175
000235      175 PUNCH 182,J,I, H(I,J)
000251      180 CONTINUE
000254          IF(CHK-2.) 200,200,190
000256      190 DO 198 I = ISEC2,ILAST
000260          IF( H(I,J))195,198,195
000264      195 PUNCH 202,J,I, H(I,J)
000300      198 CONTINUE
000303      200 CONTINUE
000306      205 IF(CHJ-2.) 260,260,210
000311      210 DO 250 J = JSEC2,JLAST
000313          DO 220 I = 1,IFRST
000314          IF( H(I,J))218,220,218
000320      218 PUNCH 223,J,I, H(I,J)
000334      220 CONTINUE
000337          IF(CHK-1.) 250,250,225
000341      225 DO 235 I = IFRS1,ISEC
000343          IF( H(I,J))230,235,230
000347      230 PUNCH 238,J,I, H(I,J)
000363      235 CONTINUE
000366          IF(CHK-2.) 250,250,240
000370      240 DO 248 I = ISEC2,ILAST
000372          IF( H(I,J))245,248,245
000376      245 PUNCH 253,J,I, H(I,J)
000412      248 CONTINUE
000415      250 CONTINUE
000420      260 RETURN
000421          END
```

```
              SUBROUTINE RHDATA
000002        COMMON/RHV/G(120),NRE(120)
000002        COMMON/ALL/ INITR,KS(8)
000002     85 FORMAT(4X,4HRHS1,6X,3HROW,I2,5X,E12.5)
000002     55 FORMAT(4X,4HRHS1,6X,3HROW,I1,6X,E12.5)
000002    115 FORMAT (4X,4HRHS1,6X,3HROW,I3,4X,E12.5)
000002        ILAST = INITR
000003        CHK = 3.
000005        IFRST = 9
000006        ISEC = 99
000007        IF(ILAST -9) 10,10,20
000012     10 IFRST = ILAST
000013        CHK = 1.
000015        GO TO 40
000016     20 IF(ILAST-99) 30,30,40
000021     30 ISEC = ILAST
000022        CHK = 2.
000024     40 IFRS1 = IFRST+1
000026        ISEC2 = ISEC+1
000030        DO 60 I = 1,IFRST
000031        IF(G(I)) 50,60,50
000033     50 PUNCH 55,I,G(I)
000043     60 CONTINUE
000046        IF(CHK-1.) 130,130,70
000050     70 DO 90 I = IFRS1,ISEC
000052        IF(G(I)) 80,90,80
000054     80 PUNCH 85,I,G(I)
000064     90 CONTINUE
000067        IF(CHK-2.) 130,130,100
000071    100 DO 120 I = ISEC2,ILAST
000073        IF(G(I)) 110,120,110
000075    110 PUNCH 115,I,G(I)
000105    120 CONTINUE
000110    130 RETURN
000111        END
```

```
                SUBROUTINE BODATA
000002          COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE
000002          JLAST=(II-1)*NU+1
000005          CHJ=3.
000007          JFRST = 9
000010          JSEC=99
000011          IF(JLAST-9)10,10,20
000014       10 JFRST=JLAST
000015          CHJ=1.
000017          GO TO 40
000020       20 IF(JLAST-99)30,30,40
000023       30 JSEC=JLAST
000024          CHJ=2.
000026       40 JFRS1=JFRST+1
000030          JSEC2=JSEC+1
000032          DO 90 J=2,JFRST
000033          PUNCH 50,J
000040       50 FORMAT(1X,2HFR,1X,4HBDS1,6X,3HCOL,I1)
000040       90 CONTINUE
000043          IF (CHJ - 1.) 160,160,100
000045      100 DO 120 J=JFRS1,JSEC
000047          PUNCH 60,J
000054       60 FORMAT(1X,2HFR,1X,4HBDS1,6X,3HCOL,I2)
000054      120 CONTINUE
000057          IF(CHJ-2.)160,160,130
000061      130 DO 150 J=JSEC2,JLAST
000063          PUNCH 70,J
000070       70 FORMAT(1X,2HFR,1X,4HBDS1,6X,3HCOL,I3)
000070      150 CONTINUE
000073      160 RETURN
000074          END
```

```
                SUBROUTINE ELEMTP
000002          COMMON/BRG/R(20,20,4),T(20,20,4)
000002          COMMON/CSS/C1(20,20),C2(1,20,20),C(1,20,20)
000002          COMMON/FOFUN/F(1,20,3)
000002          COMMON/OBJ/PX1(5,20),PX2(5,4),PX3(5,4)
000002          COMMON/ELS/C4(20,20)
000002          COMMON/PVAL/H(120,80)
000002          COMMON/RHV/G(120),NRE(120)
000002          COMMON /RWID/ ISP   ,MSP(8,2)
000002          COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE
000002          COMMON/ALL/ INITR,KS(8)
000002          COMMON/FORC/ NSETS
000002          COMMON /RWS/ NOBF,NOC,NOB,NOT
000002          DO 1 I=1,120
000004          G(I)=0.0
000005          NRE(I)=0
000006          DO 1 J=1,80
000010          H(I,J)=0.0
000014        1 CONTINUE
000020          DO 850 I = 1,NOB
000021          CALL SEARCH (I,N   ,PX1,ISH,5,20)
000025          IF (ISH .EQ.0) GO TO 600
000026          CALL SEARCH (I,NU,PX2,ISH ,5,4)
000032          IF (ISH .EQ. 0) GO TO 500
000033      400 JL = NU+1
000035          IL = II-2
000037          IX = 2*IL
000040          II1 = II-1
000041          DO 415 K = 2,II1
000043          CALL M12X3 (K,I,N,NU,PX1,R,C4)
000051          IP = INITR +(K-1)
000054          IQ = IP+IL
000056          H(IP ,1) = -1.0
000057          H(IQ,1) = 1.
000061          DO 405 J = 2,JL
000063          H(IP,J) = C4(I,J-1)
000071          H(IQ,J) = H(IP,J)
000077          H(IP,J+NU) = PX2(I,J-1)
000105          H(IQ,J+NU) = H(IP,J+NU)
000114      405 CONTINUE
000116          IF (K.EQ. 2) GO TO 415
000120          JF = K*NU+1
000123          JL1 = JL+1
000124          DO 410 J = JL1,JF
000125          H(IP,J) = H(IP-1,J-NU)
000134          H(IQ,J) = H(IP,J)
000141      410 CONTINUE
000143      415 CONTINUE
000146          DO 450 IS = 1,NSETS
000147          DO 450 K = 2,II1
000150          JK = K*NU+1
000154          IP = INITR +(IS-1)*IX+K-1
000161          IQ = IP+IL
000163          G(IP) = 0.
000164          DO 420 J = 1,NF
000165          G(IP) = G(IP) - PX3(I,J)*F(IS,K,J)
000177      420 CONTINUE
```

```
000201          DO 423 J = 1,N
000203          G(IP) = G(IP)-PX1(I,J)*C(IS,K,J)
000215      423 CONTINUE
000217          G(IQ) = G(IP)
000222          IF (ISP .EQ.0) GO TO 425
000223          NRE(IP) = 1HG
000225          NRE(IQ) = 1HL
000227          GO TO 430
000227      425 NRE(IP) = 1HL
000231          NRE(IQ) = 1HG
000233      430 IF(IS .EQ.1) GO TO 450
000235          DO 440 J = 1,JK
000237          H(IP,J) = H(IP-IX,J)
000245          H(IQ,J) = H(IQ-IX,J)
000251      440 CONTINUE
000254      450 CONTINUE
000261          GO TO 800
000262      500 JL = NU+1
000264          IL = II-1
000266          IX = 2*IL
000267          DO 515 K = 2,II
000270          CALL M12X3 (K,I,N,NU,PX1,R,C4)
000276          IP = INITR +(K-1)
000301          IQ  = IP+IL
000303          H(IP,1) = -1.0
000304          H(IQ,1) = 1.
000306          DO 505 J = 2,JL
000310          H(IP,J)= C4(I,J-1)
000316          H(IQ,J) = H(IP,J)
000324      505 CONTINUE
000326          IF (K .EQ.2) GO TO 515
000330          JF = (K-1) *NU+1
000334          JL1 = JL+1
000335          DO 510 J = JL1,JF
000336          H(IP,J) = H(IP-1,J-NU)
000345          H(IQ,J) = H(IP,J)
000352      510 CONTINUE
000354      515 CONTINUE
000357          DO 560 IS = 1,NSETS
000360          DO 560 K = 2,II
000361          JK = (K-1)*NU+1
000365          IP = INITR+(IS-1)*IX+K-1
000372          IQ = IP+IL
000374          G(IP) = 0.0
000375          DO 520 J = 1,NF
000376          G(IP)    = G(IP) -PX3(I,J)*F(IS,K-1,J)
000410      520 CONTINUE
000412          DO 523 J = 1,N
000414          G(IP) = G(IP)-PX1(I,J)*C(IS,K,J)
000426      523 CONTINUE
000430          G(IQ )= G(IP)
000433          IF (ISP .EQ.0) GO TO 530
000434          NRE(IP) = 1HG
000436          NRE(IQ) = 1HL
000440          GO TO 540
000440      530 NRE(IP) = 1HL
000442          NRE(IQ) = 1HG
000444      540 IF (IS .EQ. 1) GO TO 560
```

```
000450          H(IP,J) = H(IP-IX,J)
000456          H(IQ,J) = H(IQ-IX,J)
000462      550 CONTINUE
000465      560 CONTINUE
000472          GO TO 800
000473      600 CALL SEARCH (I,NF,PX3,ISH,5,4)
000477          IF (ISH .EQ.0) GO TO 700
000500          IL = II-1
000502          DO 630 IS = 1,NSETS
000504          DO 630 K = 1,IL
000505          IP = (IS-1) *(II-1)*2+(K-1)+1+INITR
000515          IQ = IP+IL
000517          H(IP,1) = -1.0
000521          H(IQ,1) = 1.0
000523          IF (ISP .EQ.0) GO TO 605
000524          NRE(IP) = 1HG
000525          NRE(IQ) = 1HL
000527          GO TO 610
000527      605 NRE(IP) = 1HL
000531          NRE(IQ) = 1HG
000533      610 G(IP) = 0.0
000535          DO 620 L = 1,NF
000536          G(IP) = G(IP)-PX3(I,L)*F(IS,K,L)
000550      620 CONTINUE
000552          J1 = 1+(K-1)*NU
000556          DO 635 J = 1,NU
000560          JP = J1+J
000562          H(IP,JP) = PX2(I,J)
000570          H(IQ,JP) = PX2(I,J)
000575      635 CONTINUE
000577      630 CONTINUE
000604          GO TO 800
000605      700 IL = II-1
000607          DO 720 K = 1,IL
000611          J1 = 1+(K-1)*NU
000615          I1 = INITR +(K-1)
000617          IP = I1+1
000620          IQ = IL+IP
000621          H(IP,1) = -1.0
000623          H(IQ,1) = 1.0
000625          IF (ISP .EQ.0) GO TO 705
000626          NRE(IP) = 1HG
000627          NRE(IQ) = 1HL
000631          GO TO 710
000631      705 NRE(IP) = 1HL
000633          NRE(IQ) = 1HG
000635      710 DO 715 J = 1,NU
000637          JP = J1+J
000641          H(IP,JP) = PX2(I,J)
000647          H(IQ,JP) = PX2(I,J)
000654      715 CONTINUE
000656      720 CONTINUE
000661      800 INITR = IQ
000663      850 CONTINUE
000665          RETURN
000666          END
```

```
                SUBROUTINE ELEMTQ
000002          COMMON /RWIO/ ISP   ,MSP(8,2)
000002          COMMON /RWS/ NOBF,NOC,NOB,NOT
000002          DO 399 I = 1,NOC
000004      200 IF (MSP(I,2) -1) 250,251,205
000007      205 IF (MSP(I,2) -2) 252,252,253
000013      250 CALL ELEMTR (I)
000015          CALL ELEMTS (I)
000017          GO TO 399
000020      251 CALL ELEMTR (I)
000022          GO TO 399
000023      252 CALL ELEMTS (I)
000025          GO TO 399
000026      253 CALL ELEMTT (I)
000030      399 CONTINUE
000033          RETURN
000033          END
```

```
000003          COMMON/PVAL/H(120,80)
000003          COMMON/RHV/G(120),NRE(120)
000003          COMMON/BRG/R(20,20,4),T(20,20,4)
000003          COMMON/CSS/C1(20,20),C2(1,20,20),C(1,20,20)
000003          COMMON/FOFUN/F(1,20,3)
000003          COMMON/UPLO/YU(20,5),YL(20,5),Y(20)
000003          COMMON/CONST/Y1(5,20),Y2(5,4),Y3(5,4)
000003          COMMON/ELS/C4(20,20)
000003          COMMON /RWID/ ISP   ,MSP(8,2)
000003          COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE
000003          COMMON/ALL/ INITR,KS(8)
000003          COMMON/FORC/ NSETS
000003          COMMON /RWS/ NOBF,NOC,NOB,NOT
000003          CALL SEARCH (I,N,Y1,ISH,5,20)
000007          IF (ISH .EQ. 0) GO TO 400
000011          CALL SEARCH (I,NU,Y2,ISH,5,4)
000015          IF (ISH .EQ. 0) GO TO 300
000017      200 IF (MSP (I,1) .NE. 0) GO TO 205
000021          KINIT = 2
000022          KLAST = II-1
000024          GO TO 208
000024      205 KINIT = KS(I)
000026          KLAST = KS(I)
000030          IP=INITR+1
000032          DO 213 L=2, KINIT
000033          LL=KINIT -L+2
000036          CALL M12X3(LL,I,N,NU,Y1,R,C4)
000045          J1=1+(L-2)*NU
000052          J11=J1+1
000053          J2=J1+NU
000054          DO 214 J=J11,J2
000056      214 H(IP,J)=C4(I,J-J1)
000071      213 CONTINUE
000074          J3=J2+1
000076          J4=J2+NU
000100          DO 216 J=J3,J4
000102      216 H(IP,J)=Y2(I,J-J2)
000115          GO TO 227
000116      208 JL = NU+1
000120          IL = II-2
000122          DO 220 K = KINIT ,KLAST
000124          CALL M12X3 (K,I,N,NU,Y1,R,C4)
000133          IF (MSP(I,1) .NE. 0) GO TO 209
000136          IP = INITR +(K-1)
000141          GO TO 211
000141      209 IP = INITR +1
000143      211 DO 210 J = 2,JL
000145          H(IP,J) =C4(I,J-1)
000153          H(IP,J +NU) = Y2(I,J-1)
000161      210 CONTINUE
000164          IF (MSP(I,1) .NE. 0) GO TO 220
000166          IF (K .LE. 2) GO TO 220
000170          JF = K*NU+1
000174          JL1 = JL+1
000175          DO 215 J = JL1,JF
000176          H(IP,J) = H(IP-1,J-NU)
```

```
000206      215 CONTINUE
000210      220 CONTINUE
000213      227 DO 240 IS = 1,NSETS
000215          DO 240 K = KINIT,KLAST
000217          JK = K*NU+1
000223          IF (MSP(I,1) .NE. 0) GO TO 221
000225          IP = INITR+(IS-1)*IL+K-1
000232          GO TO 222
000233      221 IP = INITR+(IS-1)+1
000236          K = KS(I)
000240          IL = 1
000241      222 G(IP) = 0.0
000243          DO 230 J = 1,NF
000244          G(IP)    = G(IP) - Y3(I,J)*F(IS,K,J)
000255      230 CONTINUE
000257          DO 233 J = 1,N
000261          G(IP) = G(IP)- Y1(I,J)*C(IS,K,J)
000272      233 CONTINUE
000274          G(IP) = G(IP)+YU(K,I)
000302          NRE(IP) = 1HL
000303          IF (IS .EQ. 1) GO TO 240
000305          DO 235 J = 1,JK
000307          H(IP,J)   =   H(IP-IL,J)
000317      235 CONTINUE
000321      240 CONTINUE
000326          GO TO 550
000327      300 IF (MSP(I,1) .NE. 0) GO TO 305
000331          KINIT = 2
000332          KLAST = II
000333          GO TO 308
000334      305 KINIT = KS(I)
000336          KLAST = KS(I)
000340          IP=INITR+1
000342          DO 313 L=2, KINIT
000343          LL=KINIT -L+2
000346          CALL M12X3(LL,I,N,NU,Y1,R,C4)
000355          J1=1+(L-2)*NU
000362          J11=J1+1
000363          J2=J1+NU
000364          DO 314 J=J11,J2
000366      314 H(IP,J)=C4(I,J-J1)
000401      313 CONTINUE
000404          GO TO 332
000404      308 JL = NU+1
000406          IL = II-1
000410          DO 320 K = KINIT ,KLAST
000412          KJ=K-1
000414          DO 401 J=1,NU
000415          C4(I,J)=0.0
000420          DO 401 L=1,N
000422          C4(I,J)=C4(I,J)+Y1(I,L)*R(KJ,L,J)
000436      401 CONTINUE
000443          IF (MSP(I,1) .NE. 0) GO TO 309
000445          IP = INITR +K-1
000447          GO TO 311
000450      309 IP = INITR +1
000452          K = KS(I)
000454      311 DO 310 J = 2,JL
```

```
000456            H(IP,J ) = C4(I,J-1)
000465       310  CONTINUE
000467            IF (MSP(I,1) .NE. 0) GO TO 320
000471            IF (K .LE. 2) GO TO 320
000473            JF = (K-1)*NU+1
000477            JL1 = JL+1
000500            DO 315 J = JL1,JF
000501            H(IP,J) = H(IP-1,J-NU)
000511       315  CONTINUE
000513       320  CONTINUE
000516       332  DO 335 IS = 1,NSETS
000520            DO 335 K = KINIT ,KLAST
000522            JK = (K-1)*NU+1
000526            IF (MSP(I,1) .NE. 0) GO TO 321
000530            IP = INITR+(IS-1)*IL+K-1
000535            GO TO 322
000536       321  IP = INITR+(IS-1)+1
000541            K = KS(I)
000543            IL = 1
000544       322  G(IP) = 0.0
000546            DO 325 J = 1,NF
000547            G(IP)    = G(IP) - Y3(I,J)*F(IS,K,J)
000560       325  CONTINUE
000562            DO 328 J = 1,N
000564            G(IP) = G(IP)- Y1(I,J)*C(IS,K,J)
000575       328  CONTINUE
000577            G(IP) = G(IP)+YU(K,I)
000605            NRE(IP) = 1HL
000606            IF (IS .EQ. 1) GO TO 335
000610            DO 330 J = 1,JK
000612            H(IP,J) = H(IP-IL,J)
000622       330  CONTINUE
000624       335  CONTINUE
000631            GO TO 550
000632       400  CALL SEARCH (I,NF,Y3,ISH,5,4)
000636            IF (ISH .EQ.0) GO TO 500
000640            IF (MSP(I,1) .NE.0) GO TO 405
000642            KINIT = 1
000643            KLAST = II-1
000645            GO TO 408
000645       405  KINIT = KS(I)
000647            KLAST = KS(I)
000651       408  IL = II-1
000653            DO 420 IS = 1,NSETS
000655            DO 420 K = KINIT,KLAST
000657            IF (MSP(I,1) .NE. 0) GO TO 409
000661            IP = (IS-1)*(II-1)+(K-1)+1+INITR
000670            GO TO 411
000671       409  IP = INITR+(IS-1)+1
000674            K = KS(I)
000676       411  NRE(IP) = 1HL
000700            G(IP) = 0.0
000701            DO 410 L = 1,NF
000703       410  G(IP) = G(IP)-Y3(I,L)*F(IS,K,L)
000716            G(IP) = G(IP)+YU(K,I)
000724            J1 = 1+(K-1)*NU
000727            DO 415 J = 1,NU
000731            JP =  J1+J
```

```
000733          H(IP,JP) = Y2(I,J)
000741      415 CONTINUE
000743      420 CONTINUE
000750          GO TO 550
000750      500 IF (MSP(I,1) .NE.0) GO TO 505
000752          KINIT = 1
000753          KLAST = II-1
000755          GO TO 508
000755      505 KINIT = KS(I)
000757          KLAST = KS(I)
000761      508 IL = II-1
000763          DO 515 K = KINIT,KLAST
000765          J1 = 1+(K-1)*NU
000771          IF (MSP(I,1) .NE. 0) GO TO 506
000773          I1 = INITR+(K-1)
000776          IP = I1+1
000777          GO TO 507
000777      506 IP = INITR +1
001001          K = KS(I)
001003      507 NRE(IP) = 1HL
001005          G(IP) = YU(K,I)
001012          DO 510 J = 1,NU
001013          JP = J1+J
001015          H(IP,JP) = Y2(I,J)
001023      510 CONTINUE
001025      515 CONTINUE
001027      550 INITR = IP
001031          RETURN
001031          END
```

```
            SUBROUTINE ELEMTS (I)
000003      COMMON/PVAL/H(120,80)
000003      COMMON/RHV/G(120),NRE(120)
000003      COMMON/ELS/C4(20,20)
000003      COMMON/BRG/R(20,20,4),T(20,20,4)
000003      COMMON/CSS/C1(20,20),C2(1,20,20),C(1,20,20)
000003      COMMON/FOFUN/F(1,20,3)
000003      COMMON/UPLO/YU(20,5),YL(20,5),Y(20)
000003      COMMON/CONST/Y1(5,20),Y2(5,4),Y3(5,4)
000003      COMMON /RWID/ ISP   ,MSP(8,2)
000003      COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE
000003      COMMON/ALL/ INITR,KS(8)
000003      COMMON/FORC/ NSETS
000003      COMMON /RHS/ NOBF,NOC,NOB,NOT
000003      CALL SEARCH (I,N,Y1,ISH,5,20)
000007      IF (ISH .EQ. 0) GO TO 400
000011      CALL SEARCH (I,NU,Y2,ISH,5,4)
000015      IF (ISH .EQ. 0) GO TO 300
000017  200 IF (MSP (I,1) .NE. 0) GO TO 205
000021      KINIT = 2
000022      KLAST = II-1
000024      GO TO 208
000024  205 KINIT = KS(I)
000026      KLAST = KS(I)
000030      IP=INITR+1
000032      DO 213 L=2, KINIT
000033      LL=KINIT -L+2
000036      CALL M12X3(LL,I,N,NU,Y1,R,C4)
000045      J1=1+(L-2)*NU
000052      J11=J1+1
000053      J2=J1+NU
000054      DO 214 J=J11,J2
000056  214 H(IP,J)=C4(I,J-J1)
000071  213 CONTINUE
000074      J3=J2+1
000076      J4=J2+NU
000100      DO 216 J=J3,J4
000102  216 H(IP,J)=Y2(I,J-J2)
000115      GO TO 227
000116  208 JL = NU+1
000120      IL = II-2
000122      DO 220 K = KINIT ,KLAST
000124      CALL M12X3 (K,I,N,NU,Y1,R,C4)
000133      IF (MSP(I,1) .NE. 0) GO TO 209
000136      IP = INITR +(K-1)
000141      GO TO 211
000141  209 IP = INITR +1
000143  211 DO 210 J = 2,JL
000145      H(IP,J) =C4(I,J-1)
000153      H(IP,J +NU) = Y2(I,J-1)
000161  210 CONTINUE
000164      IF (MSP(I,1) .NE. 0) GO TO 220
000166      IF (K .LE. 2) GO TO 220
000170      JF = K*NU+1
000174      JL1 = JL+1
000175      DO 215 J = JL1,JF
000176      H(IP,J) = H(IP-1,J-NU)
```

```
000206      215 CONTINUE
000210      220 CONTINUE
000213      227 DO 240 IS = 1,NSETS
000215          DO 240 K = KINIT,KLAST
000217          JK = K*NU+1
000223          IF (MSP(I,1) .NE. 0) GO TO 221
000225          IP = INITR+(IS-1)*IL+K-1
000232          GO TO 222
000233      221 IP = INITR+(IS-1)+1
000236          K = KS(I)
000240          IL = 1
000241      222 G(IP) = 0.0
000243          DO 230 J = 1,NF
000244          G(IP)    = G(IP)  - Y3(I,J)*F(IS,K,J)
000255      230 CONTINUE
000257          DO 233 J = 1,N
000261          G(IP) = G(IP)- Y1(I,J)*C(IS,K,J)
000272      233 CONTINUE
000274          G(IP) = G(IP)+YL(K,I)
000302          NRE(IP) = 1HG
000303          IF (IS .EQ. 1) GO TO 240
000305          DO 235 J = 1,JK
000307          H(IP,J)  =  H(IP-IL,J)
000317      235 CONTINUE
000321      240 CONTINUE
000326          GO TO 550
000327      300 IF (MSP(I,1) .NE. 0) GO TO 305
000331          KINIT = 2
000332          KLAST = II
000333          GO TO 308
000334      305 KINIT = KS(I)
000336          KLAST = KS(I)
000340          IP=INITR+1
000342          DO 313 L=2, KINIT
000343          LL=KINIT -L+2
000346          CALL M12X3(LL,I,N,NU,Y1,R,C4)
000355          J1=1+(L-2)*NU
000362          J11=J1+1
000363          J2=J1+NU
000364          DO 314 J=J11,J2
000366      314 H(IP,J)=C4(I,J-J1)
000401      313 CONTINUE
000404          GO TO 332
000404      308 JL = NU+1
000406          IL = II-1
000410          DO 320 K = KINIT ,KLAST
000412          CALL M12X3 (K,I,N,NU,Y1,R,C4)
000421          IF (MSP(I,1) .NE. 0) GO TO 309
000424          IP = INITR +K-1
000426          GO TO 311
000427      309 IP = INITR +1
000431          K = KS(I)
000433      311 DO 310 J = 2,JL
000435          H(IP,J ) = C4(I,J-1)
000444      310 CONTINUE
000446          IF (MSP(I,1) .NE. 0) GO TO 320
000450          IF (K .LE. 2) GO TO 320
000452          JF = (K-1)*NU+1
```

```
000456         JL1 = JL+1
000457         DO 315 J = JL1,JF
000460         H(IP,J) = H(IP-1,J-NU)
000470     315 CONTINUE
000472     320 CONTINUE
000475     332 DO 335 IS = 1,NSETS
000477         DO 335 K = KINIT ,KLAST
000501         JK = (K-1)*NU+1
000505         IF (MSP(I,1) .NE. 0) GO TO 321
000507         IP = INITR+(IS-1)*IL+K-1
000514         GO TO 322
000515     321 IP = INITR+(IS-1)+1
000520         K = KS(I)
000522         IL = 1
000523     322 G(IP) = 0.0
000525         DO 325 J = 1,NF
000526         G(IP)   = G(IP) - Y3(I,J)*F(IS,K,J)
000537     325 CONTINUE
000541         DO 328 J = 1,N
000543         G(IP) = G(IP)- Y1(I,J)*C(IS,K,J)
000554     328 CONTINUE
000556         G(IP) = G(IP)+YL(K,I)
000564         NRE(IP) = 1HG
000565         IF (IS .EQ. 1) GO TO 335
000567         DO 330 J = 1,JK
000571         H(IP,J) = H(IP-IL,J)
000601     330 CONTINUE
000603     335 CONTINUE
000610         GO TO 550
000611     400 CALL SEARCH (I,NF,Y3,ISH,5,4)
000615         IF (ISH .EQ.0) GO TO 500
000617         IF (MSP(I,1) .NE.0) GO TO 405
000621         KINIT = 1
000622         KLAST = II-1
000624         GO TO 408
000624     405 KINIT = KS(I)
000626         KLAST = KS(I)
000630     408 IL = II-1
000632         DO 420 IS = 1,NSETS
000634         DO 420 K = KINIT,KLAST
000636         IF (MSP(I,1) .NE. 0) GO TO 409
000640         IP = (IS-1)*(II-1)+(K-1)+1+INITR
000647         GO TO 411
000650     409 IP = INITR+(IS-1)+1
000653         K = KS(I)
000655     411 NRE(IP) = 1HG
000657         G(IP) = 0.0
000660         DO 410 L = 1,NF
000662     410 G(IP) = G(IP)-Y3(I,L)*F(IS,K,L)
000675         G(IP) = G(IP)+YL(K,I)
000703         J1 = 1+(K-1)*NU
000706         DO 415 J = 1,NU
000710         JP =  J1+J
000712         H(IP,JP) = Y2(I,J)
000720     415 CONTINUE
000722     420 CONTINUE
000727         GO TO 550
000727     500 IF (MSP(I,1) .NE.0) GO TO 505
```

```
000731          KINIT = 1
000732          KLAST = II-1
000734          GO TO 508
000734      505 KINIT = KS(I)
000736          KLAST = KS(I)
000740      508 IL = II-1
000742          DO 515 K = KINIT,KLAST
000744          J1 = 1+(K-1)*NU
000750          IF (MSP(I,1) .NE. 0) GO TO 506
000752          I1 = INITR+(K-1)
000755          IP = I1+1
000756          GO TO 507
000756      506 IP = INITR +1
000760          K = KS(I)
000762      507 NRE(IP) = 1HG
000764          G(IP) = YL(K,I)
000771          DO 510 J = 1,NU
000772          JP = J1+J
000774          H(IP,JP) = Y2(I,J)
001002      510 CONTINUE
001004      515 CONTINUE
001006      550 INITR = IP
001010          RETURN
001010          END
```

```
000003         COMMON/RHV/G(120),NRE(120)
000003         COMMON/PVAL/H(120,80)
000003         COMMON/ELS/C4(20,20)
000003         COMMON/BRG/R(20,20,4),T(20,20,4)
000003         COMMON/CSS/C1(20,20),C2(1,20,20),C(1,20,20)
000003         COMMON/FOFUN/F(1,20,3)
000003         COMMON/UPLO/YU(20,5),YL(20,5),Y(20)
000003         COMMON/CONST/Y1(5,20),Y2(5,4),Y3(5,4)
000003         COMMON /RWID/ ISP   ,MSP(8,2)
000003         COMMON /INPT/ II,N,M,NF,NU,KI,KJ,NDF,MODE
000003         COMMON/ALL/ INITR,KS(8)
000003         COMMON/FORC/ NSETS
000003         COMMON /RWS/ NOBF,NOC,NOB,NOT
000003         CALL SEARCH (I,N,Y1,ISH,5,20)
000007         IF (ISH .EQ. 0) GO TO 400
000011         CALL SEARCH (I,NU,Y2,ISH,5,4)
000015         IF (ISH .EQ. 0) GO TO 300
000017     200 IF (MSP (I,1) .NE. 0) GO TO 205
000021         KINIT = 2
000022         KLAST = II-1
000024         GO TO 208
000024     205 KINIT = KS(I)
000026         KLAST = KS(I)
000030         IP=INITR+1
000032         DO 213 L=2, KINIT
000033         LL=KINIT -L+2
000036         CALL M12X3(LL,I,N,NU,Y1,R,C4)
000045         J1=1+(L-2)*NU
000052         J11=J1+1
000053         J2=J1+NU
000054         DO 214 J=J11,J2
000056     214 H(IP,J)=C4(I,J-J1)
000071     213 CONTINUE
000074         J3=J2+1
000076         J4=J2+NU
000100         DO 216 J=J3,J4
000102     216 H(IP,J)=Y2(I,J-J2)
000115         GO TO 227
000116     208 JL = NU+1
000120         IL = II-2
000122         DO 220 K = KINIT ,KLAST
000124         CALL M12X3 (K,I,N,NU,Y1,R,C4)
000133         IF (MSP(I,1) .NE. 0) GO TO 209
000136         IP = INITR +(K-1)
000141         GO TO 211
000141     209 IP = INITR +1
000143     211 DO 210 J = 2,JL
000145         H(IP,J) =C4(I,J-1)
000153         H(IP,J +NU) = Y2(I,J-1)
000161     210 CONTINUE
000164         IF (MSP(I,1) .NE. 0) GO TO 220
000166         IF (K .LE. 2) GO TO 220
000170         JF = K*NU+1
000174         JL1 = JL+1
000175         DO 215 J = JL1,JF
000176         H(IP,J) = H(IP-1,J-NU)
```

```
000206      215 CONTINUE
000210      220 CONTINUE
000213      227 DO 240 IS = 1,NSETS
000215          DO 240 K = KINIT,KLAST
000217          JK = K*NU+1
000223          IF (MSP(I,1) .NE. 0) GO TO 221
000225          IP = INITR+(IS-1)*IL+K-1
000232          GO TO 222
000233      221 IP = INITR+(IS-1)+1
000236          K = KS(I)
000240          IL = 1
000241      222 G(IP) = 0.0
000243          DO 230 J = 1,NF
000244          G(IP)   = G(IP) - Y3(I,J)*F(IS,K,J)
000255      230 CONTINUE
000257          DO 233 J = 1,N
000261          G(IP) = G(IP)- Y1(I,J)*C(IS,K,J)
000272      233 CONTINUE
000274          G(IP) = G(IP)+YU(K,I)
000302          IF (IS .EQ. 1) GO TO 240
000304          NRE(IP) = 1HE
000305          DO 235 J = 1,JK
000307          H(IP,J)  =  H(IP-IL,J)
000317      235 CONTINUE
000321      240 CONTINUE
000326          GO TO 550
000327      300 IF (MSP(I,1) .NE. 0) GO TO 305
000331          KINIT = 2
000332          KLAST = II
000333          GO TO 308
000334      305 KINIT = KS(I)
000336          KLAST = KS(I)
000340          IP=INITR+1
000342          DO 313 L=2, KINIT
000343          LL=KINIT -L+2
000346          CALL M12X3(LL,I,N,NU,Y1,R,C4)
000355          J1=1+(L-2)*NU
000362          J11=J1+1
000363          J2=J1+NU
000364          DO 314 J=J11,J2
000366      314 H(IP,J)=C4(I,J-J1)
000401      313 CONTINUE
000404          GO TO 332
000404      308 JL = NU+1
000406          IL = II-1
000410          DO 320 K = KINIT ,KLAST
000412          CALL M12X3 (K,I,N,NU,Y1,R,C4)
000421          IF (MSP(I,1) .NE. 0) GO TO 309
000424          IP = INITR +K-1
000426          GO TO 311
000427      309 IP = INITR +1
000431          K = KS(I)
000433      311 DO 310 J = 2,JL
000435          H(IP,J ) = C4(I,J-1)
000444      310 CONTINUE
000446          IF (MSP(I,1) .NE. 0) GO TO 320
000450          IF (K .LE. 2) GO TO 320
000452          JF = (K-1)*NU+1
```

```
000460          H(IP,J) = H(IP-1,J-NU)
000470      315 CONTINUE
000472      320 CONTINUE
000475      332 DO 335 IS = 1,NSETS
000477          DO 335 K = KINIT ,KLAST
000501          JK = (K-1)*NU+1
000505          IF (MSP(I,1) .NE. 0) GO TO 321
000507          IP = INITR+(IS-1)*IL+K-1
000514          GO TO 322
000515      321 IP = INITR+(IS-1)+1
000520          K = KS(I)
000522          IL = 1
000523      322 G(IP) = 0.0
000525          DO 325 J = 1,NF
000526          G(IP)    = G(IP) - Y3(I,J)*F(IS,K,J)
000537      325 CONTINUE
000541          DO 328 J = 1,N
000543          G(IP) = G(IP)- Y1(I,J)*C(IS,K,J)
000554      328 CONTINUE
000556          G(IP) = G(IP)+YU(K,I)
000564          NRE(IP) = 1HE
000565          IF (IS .EQ. 1) GO TO 335
000567          DO 330 J = 1,JK
000571          H(IP,J) = H(IP-IL,J)
000601      330 CONTINUE
000603      335 CONTINUE
000610          GO TO 550
000611      400 CALL SEARCH (I,NF,Y3,ISH,5,4)
000615          IF (ISH .EQ.0) GO TO 500
000617          IF (MSP(I,1) .NE.0) GO TO 405
000621          KINIT = 1
000622          KLAST = II-1
000624          GO TO 408
000624      405 KINIT = KS(I)
000626          KLAST = KS(I)
000630      408 IL = II-1
000632          DO 420 IS = 1,NSETS
000634          DO 420 K = KINIT,KLAST
000636          IF (MSP(I,1) .NE. 0) GO TO 409
000640          IP = (IS-1)*(II-1)+(K-1)+1+INITR
000647          GO TO 411
000650      409 IP = INITR+(IS-1)+1
000653          K = KS(I)
000655      411 NRE(IP) = 1HE
000657          G(IP) = 0.0
000660          DO 410 L = 1,NF
000662      410 G(IP) = G(IP)-Y3(I,L)*F(IS,K,L)
000675          G(IP) = G(IP)+YU(K,I)
000703          J1 = 1+(K-1)*NU
000706          DO 415 J = 1,NU
000710          JP =  J1+J
000712          H(IP,JP) = Y2(I,J)
000720      415 CONTINUE
000722      420 CONTINUE
000727          GO TO 550
000727      500 IF (MSP(I,1) .NE.0) GO TO 505
```

```
000731         KINIT = 1
000732         KLAST = II-1
000734         GO TO 508
000734     505 KINIT = KS(I)
000736         KLAST = KS(I)
000740     508 IL = II-1
000742         DO 515 K = KINIT,KLAST
000744         J1 = 1+(K-1)*NU
000750         IF (MSP(I,1) .NE. 0) GO TO 506
000752         I1 = INITR+(K-1)
000755         IP = I1+1
000756         GO TO 507
000756     506 IP = INITR +1
000760         K = KS(I)
000762     507 NRE(IP) = 1HE
000764         G(IP) = YU(K,I)
000771         DO 510 J = 1,NU
000772         JP = J1+J
000774         H(IP,JP) = Y2(I,J)
001002     510 CONTINUE
001004     515 CONTINUE
001006     550 INITR = IP
001010         RETURN
001010         END
```

```
000011        DIMENSION A(NI,NJ)
000011        DO 500 J = 1,NX
000012        IF (A(IX,J)) 510,500,510
000016    500 CONTINUE
000021        ISH = 0
000021        RETURN
000022    510 ISH = 1
000023        RETURN
000024        END
```

```
                SUBROUTINE M12X3 ( KI,IX,K,M,A,B,C)
000012          DIMENSION A(5,20),B(20,20,4),C(20,20)
000012          KJ = KI-1
000014          DO 400 J = 1,M
000015          C(IX,J) = 0.0
000021          DO 400 L = 1,K
000022          C(IX,J) = C(IX,J)+A(IX,L)*B(KJ,L,J)
000041      400 CONTINUE
000045          RETURN
000045          END
```

```
000011         DIMENSION A(10,10),B(10,10),C(10,10)
000011         DO 200 I = 1,N
000012         DO 200 J = 1,M
000013         C(I,J) = 0.0
000016         DO 200 L = 1,K
000020     200 C(I,J) = C(I,J) +A(I,L) *B(L,J)
000043         RETURN
000043         END
```

```
                    SUBROUTINE  MEXAP
000002              COMMON/MXAP/AEI(20,20),P(20,20)
000002              COMMON/ABD/A(20,20),B(20,4),D(20,4)
000002              COMMON/INPT/II,N,MQ,NF,NU,KI,KJ,NDF,MODE,IZ
000002              COMMON/T/TM
000002              DIMENSION X(20,20),XT(20,20),Y(20,20),YT(20,20)
000002              DIMENSION U(6),V(4),W(4)
000002              T=TM
000004              Q=0.28989933
000005              U(1)=0.2499986842
000007              U(2)=0.0312575832
000010              U(3)=0.0025913712
000012              U(4)=0.0001715620
000013              U(5)=0.0000054302
000015              U(6)=0.0000006906
000016              V(1)=-2.21781431
000020              V(2)=3.33131912
000021              V(3)=-1.62781495
000023              V(4)=0.51431014
000024              W(1)=0.28905386
000026              W(2)=-0.33240494
000027              W(3)=0.45548498
000031              W(4)=0.58785466
000032              DO 1 I=1,N
000034              DO 1 J=1,N
000035              X(I,J)=0.0
000041            1 CONTINUE
000045              DO 20 I=1,N
000047              DO 10 J=1,N
000050               P(I,J)=0.0
000053               Y(I,J)=0.0
000056              X(I,J)=-T*A(I,J)
000064           10 CONTINUE
000066                P(I,I)=1.0
000072                Y(I,I)=1.0
000075           20 CONTINUE
000077              DO 60 M=1,6
000100              DO    40 I=1,N
000101              DO    40 J=1,N
000102              XT(I,J)=0.0
000105              DO    30 L=1,N
000107              XT(I,J)=XT(I,J)+X(I,L)*Y(L,J)
000121           30 CONTINUE
000123                P(I,J)=  P(I,J)+XT(I,J)*U(M)
000131           40 CONTINUE
000136              DO    50 I=1,N
000137              DO    50 J=1,N
000140              Y(I,J)=XT(I,J)
000147           50 CONTINUE
000153           60 CONTINUE
000155              DO    70 I=1,N
000157              DO    70 J=1,N
000160              XT(I,J)=0.0
000163              DO 70 L=1,N
000165              XT(I,J)=XT(I,J)+ P(I,L)*P(L,J)
000177           70 CONTINUE
000205              DO    80 I=1,N
```

```
000213          DO 80 L=1,N
000215          Y(I,J)=Y(I,J)+XT(I,L)*XT(L,J)
000227       80 CONTINUE
000235          CALL MIV(Y,N,P,20)
000240          DO  110 I=1,N
000242          DO   90 J=1,N
000243          XT(I,J)=Q*X(I,J)
000253       90 CONTINUE
000255          XT(I,I)=1.0+XT(I,I)
000261      110 CONTINUE
000263          CALL MIV(XT,N,X,20)
000266          DO  130 I=1,N
000270          DO  120 J=1,N
000271          YT(I,J)=0.0
000274           Y(I,J)=0.0
000277          AEI(I,J)=0.0
000302      120 CONTINUE
000304          AEI(I,I)=1.0
000310            Y(I,I)=1.0
000313      130 CONTINUE
000315          DO 170 M=1,4
000316          DO  150 I=1,N
000317          DO  150 J=1,N
000320          XT(I,J)=0.0
000323          DO  140 L=1,N
000325          XT(I,J)=XT(I,J)+X(I,L)*Y(L,J)
000337      140 CONTINUE
000341          AEI(I,J)=AEI(I,J)+XT(I,J)*V(M)
000347           YT(I,J)= YT(I,J)+XT(I,J)*W(M)
000355      150 CONTINUE
000362          DO  160 I=1,N
000363          DO  160 J=1,N
000364          Y(I,J)=XT(I,J)
000373      160 CONTINUE
000377      170 CONTINUE
000401          CALL MIV(AEI,N,XT,20)
000404          DO  190 I=1,N
000406          DO  190 J=1,N
000407          AEI(I,J)=0.0
000412          DO  180 L=1,N
000414          AEI(I,J)=AEI(I,J)+XT(I,L)*YT(L,J)
000426      180 CONTINUE
000430          AEI(I,J)=T*AEI(I,J)
000434      190 CONTINUE
000440          RETURN
000440          END
```

```
                 SUBROUTINE  MIV(EM,N,EMI,N1)
000007           DIMENSION EM(N1,N1),EMI(N1,N1)
000007           DO 10 I = 1,N
000010           DO 10  J = 1,N
000011        10 EMI (I,J) = EM(I,J)
000025           DO 1 I = 1,N
000027           X = EMI(I,I)
000033           EMI(I,I) = 1.0
000037           DO 2 J = 1,N
000041         2 EMI(I,J) = EMI(I,J)/X
000050           DO 1 K = 1,N
000051           IF (K .EQ.I) GO TO 1
000053           X = EMI(K,I)
000056           EMI(K,I) = 0.0
000061           DO 4 J = 1,N
000063         4 EMI(K,J) = EMI(K,J) -X *EMI(I,J)
000100         1 CONTINUE
000105           RETURN
000105           END
```

B. Program PSTPROC

```
000003          COMMON TM
000003          COMMON/YSS/ Y(20)
000003          COMMON/ALL/ INITR
000003          COMMON /INPT/ II,N,NF,NU,NDF,NSETS,NOT
000003          COMMON/CSS/C1(50,20),C2(50,20),C(1,50,20)
000003          COMMON/BRG/R(50,20,4)
000003          COMMON/FOFUN/F(1,50,4)
000003          COMMON/TAB/Q1(20,20),Q2(20,4),Q3(20,4),ISET(20)
000003           DIMENSION U(50,8)
000003           DIMENSION Z(8),TB(50),PHI(21),ITR(8)
000003      402 FORMAT(6X,5HWHERE/)
000003      401 FORMAT(3X,21H TB= Q1*S+ Q2*U+ Q3*F/)
000003      403 FORMAT(8X,3HQ1=,20F6.2)
000003      404 FORMAT(8X,3HQ2=,20F6.2)
000003      405 FORMAT(8X,3HQ3=,20F6.2)
000003      406 FORMAT(//40X,13HTIME INTERVAL,12X,2HTB/)
000003      400 FORMAT(37X,F7.3,4H TO ,F7.3, 6X,F12.6)
000003     1000 FORMAT(44X,F18.8)
000003       25 FORMAT (F12.6)
000003       30 FORMAT(6E12.6)
000003       31 FORMAT(14I5)
000003       32 FORMAT(10I5,F10.6)
000003          IPB=0
000004          READ(5,304) NPB
000012      304 FORMAT (I2)
000012      303 READ (5,32) II,N,NF,NU,NDF,NSETS,NOT,IZ,ITD,KSWT,TM
000044          IPB=IPB+1
000046          IU=II-1
000047          II1 = II-1
000050           INM1=(+3HSOL)
000052          INM2=(+3HEND)
000053          DO 20 I = 1,NOT
000055       20 READ (5,30) (Q1(I,J),J = 1,N)
000074          DO  40 I = 1,NOT
000075       40 READ(5,30) (Q2(I,J) ,J=1,NU)
000114          DO 50 I = 1,NOT
000115       50 READ (5,30) (Q3(I,J),J= 1,NF)
000134          READ (5,30) (Y(I),I = 1,IZ)
000146          DO 80 K = 1,II1
000150          DO 80 I = 1,N
000151       80 READ (5,30) (R(K,I,J),J = 1,NU)
000174          DO 90 IS = 1,NSETS
000176          DO 90 K = 1,II1
000177       90 READ (5,30) (F(IS,K,I) ,I = 1,NF)
000221          DO 100 IS = 1,NSETS
000223          DO 100 K = 2,II
000224      100 READ (5,30) (C(IS,K,I),I = 1,N)
000246          READ (5,31) (ISET(I),I = 1,NOT)
000261          READ (5,31) (ITR(I),I = 1,NOT)
000274          XX=2.0
000276          I3=0
000277          IDF=100
000300          IOLD=0
000301     1001 READ(5,191)NAME
000307      191 FORMAT(A3)
000307          IF(NAME.EQ.INM1)GO TO 111
```

```
000311           IF (NAME.EQ.INM2)GO TO 450
000313       111 DO 110 K=1,II1
000315           DO 110 J=1,NU
000316       110 U(K,J)=0.0
000327           WRITE (6,123)
000332       123 FORMAT (2X,11HHERE IS 110)
000332           READ(5,1000)T
000340           PHI(I3)=T
000342           DO 4000 I=1,IU
000344           DO 4000 J=1,NU
000345           READ(5,1000)T
000352           U(I,J)=T
000357      4000 CONTINUE
000363           WRITE (6,321)
000367       321 FORMAT (2X,12HHERE IS 1100)
000367      1100 DO 420 I=1,NOT
000371           WRITE (6,120)
000374       120 FORMAT (1H1)
000374           X1 = 0
000375           X2 = 0
000376           X3 = 0
000377           X4 = 0
000400           IS = ISET(I)
000402           CALL SEARCH (I,N,Q1,IQ1)
000405           CALL SEARCH (I,NU,Q2,IQ2)
000410           IF (IQ1 .EQ. 0) GO TO 275
000411       200 IF (IQ2 .EQ.0) GO TO 250
000412           KINIT = 2
000413           KLAST = II-1
000415           GO TO 300
000416       250 KINIT = 2
000417           KLAST = II
000421           GO TO 300
000421       275 KINIT = 1
000422           KLAST = II-1
000424       300 WRITE(6,401)
000430           WRITE(6,402)
000434           WRITE(6,403)(Q1(I,J),J=1,N)
000451           WRITE(6,404)(Q2(I,J),J=1,NU)
000466           WRITE(6,405)(Q3(I,J),J=1,NF)
000503           WRITE(6,406)
000507           DO 410 K = KINIT,KLAST
000511           IF (IQ1 .EQ. 0) GO TO 325
000512           X1 = 0
000513           DO 310 J = 1,N
000514           X1 = Q1(I,J) *C(IS,K,J) +X1
000525       310 CONTINUE
000527           K1 = K-1
000531           DO 320 J = 1,N
000533           Z(J) = 0.0
000534           DO 320 L = 1,K1
000536           DO 320 LL = 1,NU
000537           Z(J) = Z(J) +R(K-L,J,LL)*U(L,LL)
000554       320 CONTINUE
000563           X2 = 0
000564           DO 330 J = 1,N
000565           X2 = Q1(I,J)*Z(J)+X2
000573       330 CONTINUE
```

```
000576          X3 = 0
000577          DO 340 J = 1,NU
000601          X3 = X3 +Q2(I,J) *U(K,J)
000613      340 CONTINUE
000615      350 X4 = 0
000616          DO 360 J = 1,NF
000620          X4 = X4+ Q3(I,J)*F(IS,K,J)
000632      360 CONTINUE
000634          TB(K) = X1+X2+X3+X4
000641          FK=K
000642          TK1=(FK-1.0)*TM
000645          TK2=TK1+TM
000646          WRITE (6,400) TK1,TK2,TB(K)
000660      410 CONTINUE
000663          IF (ITR(I) .EQ. 0) GO TO 420
000665          CALL TRAJ (KINIT,KLAST,TB)
000667      420 CONTINUE
000672          GOTO 1001
        C       GO TO 1001
000672      450 IF (ITD .EQ. 0) GO TO 999
000673          WRITE (6,453)
000677      453 FORMAT (1H1)
000677          DO 455 I = 1,IZ
000701      455 WRITE (6,460) PHI(I),Y(I)
000713      460 FORMAT(1X,20HOBJECTIVE FUNCTION =,F12.6,5X,7HBOUND =,F12.6)
000713          CALL TRADE(PHI,Y,IZ,KSWT)
000716      999 IF (IPB .LT. NPB) GO TO 303
000721          STOP
000723          END
```

```
              SUBROUTINE TRADE (X,Y,NOC,KSWT)
000007         DIMENSION A(50),B(50),X(50),Y(50),SCL(50)
000007        WRITE (6,10)
000012     10 FORMAT (1H1,14HTRADE OFF PLOT/11X,18HOBJECTIVE FUNCTION)
000012        DO 20 J = 1,NOC
000016        B(J) = X(J)
000020     20 A(J) = Y(J)
000024        DO 68 J = 1,NOC
000026        K = NOC-J
000027        DO 68 I = 1,K
000031        IF (A(I) -A(I+1)) 40,40,30
000034     30 TEMP = A(I)
000036        A(I) = A(I+1)
000040        A(I+1) = TEMP
000042     40 IF(B(I)-B(I+1)) 60,60,50
000046     50 TEMP = B(I)
000050        B(I) = B(I+1)
000052        B(I+1) = TEMP
000054     60 IF (X(I) -X(I+1)) 65,68,68
000060     65 TEMP = X(I)
000062        X(I) = X(I+1)
000065        X(I+1) = TEMP
000066        TEMP = Y(I)
000070        Y(I) = Y(I+1)
000072        Y(I+1) = TEMP
000074     68 CONTINUE
000101        XMAXO = B(NOC)
000102        XMAXC = A(NOC)
000103        IF (KSWT .EQ. 0) GO TO 80
000105        XMINO = B(1)
000106        XMINC = A(1)
000107        XMINO = INT(XMINO)
000111        XMINC = INT(XMINC)
000113        GO TO 82
000114     80 XMINO = 0.
000115        XMINC = 0.
000116     82 RNGO = XMAXO-XMINO
000120        RNGC = XMAXC -XMINC
000122        IF (RNGO .LT. 7.) GO TO 84
000125        XINTO = INT(RNGO/8.+.9)
000130        XMAXO = XINTO*8.+ XMINO+1.0
000134        GO TO 86
000135     84 XINTO = RNGO/8.
000137        XMAXO = XINTO*8.+XMINO
000142     86 IF (RNGC .LT. 7.) GO TO 88
000145        XINTC = INT (RNGC/8.+.9)
000150        XMAXC = XINTC*8.+ XMINC
000153        GO TO 90
000154     88 XINTC = RNGC/8.
000156        XMAXC = XINTC*8.+MINC
000161     90 XINT6 = XINTO/6.
000163    100 K3 = 0
000164        J  = 1
000165        SC = 0.
000166        DO 698 K = 1,48
000170        AK = K
000171        XMAXU = XMAXO-(XINT6*(AK-1.))
```

```
000175          XMAXL = XMAXO-(XINT6*AK)
000200          IF (J .GT. NOC) GO TO 140
000203          IF (X(J) .LE. XMAXU .AND. X(J) .GT. XMAXL) GO TO 110
000215          GO TO 140
000215      110 DO 460 L = 1,9
000217          AL =  L
000220          XMIN1 = XMINC +(XINTC*(AL-1.))
000224          XMIN2 = XMINC + (XINTC*AL)
000227          IF (Y(J) .GE. XMIN1 .AND. Y(J) .LT. XMIN2) GO TO 120
000241          GO TO 460
000241      120 XINT10 = XINTC/10.
000243      130 DO 450 L1 = 1,10
000245          AL1 = L1
000246          XMIN4 = XMIN1 +(XINT10*(AL1-1.))
000252          XMIN5 = XMIN1+(XINT10*AL1)
000255          IF (Y(J) .GE. XMIN4 .AND. Y(J) .LT. XMIN5) GO TO 190
000267          GO TO 450
000267      140 K3 = K3+1
000271          IF (K3-6) 160,160,150
000273      150 K3 = 1
000274      160 IF (K3-1) 180,170,180
000276      170 SC = SC+1.
000300          XMAX2 = XMAXO-(XINTO*(SC-1.))
000304          WRITE (6,175) XMAX2
000311      175 FORMAT (10X,F10.3,1H+)
000311          GO TO 698
000314      180 WRITE (6,185)
000320      185 FORMAT (20X,1H$)
000320          GO TO 698
000323      190 K3 = K3+1
000325          IF (K3-6) 195,195,193
000327      193 K3 = 1
000330      195 IF (K3-1) 200,210,200
000332      200 GO TO (401,402,403,404,405,406,407,408,580),L
000347      210 SC = SC+1
000351          XMAX2 = XMAXO-(XINTO*(SC-1.))
000355          GO TO ( 421,422,423,424,425,426,427,428,680),L
000372      401 GO TO  (500,501,502,503,504,505,506,507,508,509),L1
000410      402 GO TO  (510,511,512,513,514,515,516,517,518,519),L1
000426      403 GO TO  (520,521,522,523,524,525,526,527,528,529),L1
000444      404 GO TO  (530,531,532,533,534,535,536,537,538,539),L1
000462      405 GO TO  (540,541,542,543,544,545,546,547,548,549),L1
000500      406 GO TO  (550,551,552,553,554,555,556,557,558,559),L1
000516      407 GO TO  (560,561,562,563,564,565,566,567,568,569),L1
000534      408 GO TO  (570,571,572,573,574,575,576,577,578,579),L1
000552      421 GO TO (600,601,602,603,604,605,606,607,608,609),L1
000570      422 GO TO (610,611,612,613,614,615,616,617,618,619),L1
000606      423 GO TO (620,621,622,623,624,625,626,627,628,629),L1
000624      424 GO TO (630,631,632,633,634,635,636,637,638,639),L1
000642      425 GO TO (640,641,642,643,644,645,646,647,648,649),L1
000660      426 GO TO (650,651,652,653,654,655,656,657,658,659),L1
000676      427 GO TO (660,661,662,663,664,665,666,667,668,669),L1
000714      428 GO TO (670,671,672,673,674,675,676,677,678,679),L1
000732      450 CONTINUE
000734      460 CONTINUE
000736      500 WRITE (6,700)
000742      700 FORMAT (20X,1H*)
000742          GO TO 692
```

```
000745      501 WRITE (6,701)
000751      701 FORMAT (20X,1H≤,    1H*)
000751          GO TO 692
000754      502 WRITE (6,702)
000760      702 FORMAT (20X,1H≤, 1X,1H*)
000760          GO TO 692
000763      503 WRITE (6,703)
000767      703 FORMAT (20X,1H≤, 2X,1H*)
000767          GO TO 692
000772      504 WRITE (6,704)
000776      704 FORMAT (20X,1H≤, 3X,1H*)
000776          GO TO 692
001001      505 WRITE (6,705)
001005      705 FORMAT (20X,1H≤, 4X,1H*)
001005          GO TO 692
001010      506 WRITE (6,706)
001014      706 FORMAT (20X,1H≤, 5X,1H*)
001014          GO TO 692
001017      507 WRITE (6,707)
001023      707 FORMAT (20X,1H≤, 6X,1H*)
001023          GO TO 692
001026      508 WRITE (6,708)
001032      708 FORMAT (20X,1H≤, 7X,1H*)
001032          GO TO 692
001035      509 WRITE (6,709)
001041      709 FORMAT (20X,1H≤, 8X,1H*)
001041          GO TO 692
001044      510 WRITE (6,710)
001050      710 FORMAT (20X,1H≤, 9X,1H*)
001050          GO TO 692
001053      511 WRITE (6,711)
001057      711 FORMAT (20X,1H≤,10X,1H*)
001057          GO TO 692
001062      512 WRITE (6,712)
001066      712 FORMAT (20X,1H≤,11X,1H*)
001066          GO TO 692
001071      513 WRITE (6,713)
001075      713 FORMAT (20X,1H≤,12X,1H*)
001075          GO TO 692
001100      514 WRITE (6,714)
001104      714 FORMAT (20X,1H≤,13X,1H*)
001104          GO TO 692
001107      515 WRITE (6,715)
001113      715 FORMAT (20X,1H≤,14X,1H*)
001113          GO TO 692
001116      516 WRITE (6,716)
001122      716 FORMAT (20X,1H≤,15X,1H*)
001122          GO TO 692
001125      517 WRITE (6,717)
001131      717 FORMAT (20X,1H≤,16X,1H*)
001131          GO TO 692
001134      518 WRITE (6,718)
001140      718 FORMAT (20X,1H≤,17X,1H*)
001140          GO TO 692
001143      519 WRITE (6,719)
001147      719 FORMAT (20X,1H≤,18X,1H*)
001147          GO TO 692
001152      520 WRITE (6,720)
```

```
001156          GO TO 692
001161      521 WRITE (6,721)
001165      721 FORMAT (20X,1H≤,20X,1H*)
001165          GO TO 692
001170      522 WRITE (6,722)
001174      722 FORMAT (20X,1H≤,21X,1H*)
001174          GO TO 692
001177      523 WRITE (6,723)
001203      723 FORMAT (20X,1H≤,22X,1H*)
001203          GO TO 692
001206      524 WRITE (6,724)
001212      724 FORMAT (20X,1H≤,23X,1H*)
001212          GO TO 692
001215      525 WRITE (6,725)
001221      725 FORMAT (20X,1H≤,24X,1H*)
001221          GO TO 692
001224      526 WRITE (6,726)
001230      726 FORMAT (20X,1H≤,25X,1H*)
001230          GO TO 692
001233      527 WRITE (6,727)
001237      727 FORMAT (20X,1H≤,26X,1H*)
001237          GO TO 692
001242      528 WRITE (6,728)
001246      728 FORMAT (20X,1H≤,27X,1H*)
001246          GO TO 692
001251      529 WRITE (6,729)
001255      729 FORMAT (20X,1H≤,28X,1H*)
001255          GO TO 692
001260      530 WRITE (6,730)
001264      730 FORMAT (20X,1H≤,29X,1H*)
001264          GO TO 692
001267      531 WRITE (6,731)
001273      731 FORMAT (20X,1H≤,30X,1H*)
001273          GO TO 692
001276      532 WRITE (6,732)
001302      732 FORMAT (20X,1H≤,31X,1H*)
001302          GO TO 692
001305      533 WRITE (6,733)
001311      733 FORMAT (20X,1H≤,32X,1H*)
001311          GO TO 692
001314      534 WRITE (6,734)
001320      734 FORMAT (20X,1H≤,33X,1H*)
001320          GO TO 692
001323      535 WRITE (6,735)
001327      735 FORMAT (20X,1H≤,34X,1H*)
001327          GO TO 692
001332      536 WRITE (6,736)
001336      736 FORMAT (20X,1H≤,35X,1H*)
001336          GO TO 692
001341      537 WRITE (6,737)
001345      737 FORMAT (20X,1H≤,36X,1H*)
001345          GO TO 692
001350      538 WRITE (6,738)
001354      738 FORMAT (20X,1H≤,37X,1H*)
001354          GO TO 692
001357      539 WRITE (6,739)
001363      739 FORMAT (20X,1H≤,38X,1H*)
```

```
001363          GO TO 692
001366     540  WRITE (6,740)
001372     740  FORMAT (20X,1H$,39X,1H*)
001372          GO TO 692
001375     541  WRITE (6,741)
001401     741  FORMAT (20X,1H$,40X,1H*)
001401          GO TO 692
001404     542  WRITE (6,742)
001410     742  FORMAT (20X,1H$,41X,1H*)
001410          GO TO 692
001413     543  WRITE (6,743)
001417     743  FORMAT (20X,1H$,42X,1H*)
001417          GO TO 692
001422     544  WRITE (6,744)
001426     744  FORMAT (20X,1H$,43X,1H*)
001426          GO TO 692
001431     545  WRITE (6,745)
001435     745  FORMAT (20X,1H$,44X,1H*)
001435          GO TO 692
001440     546  WRITE (6,746)
001444     746  FORMAT (20X,1H$,45X,1H*)
001444          GO TO 692
001447     547  WRITE (6,747)
001453     747  FORMAT (20X,1H$,46X,1H*)
001453          GO TO 692
001456     548  WRITE (6,748)
001462     748  FORMAT (20X,1H$,47X,1H*)
001462          GO TO 692
001465     549  WRITE (6,749)
001471     749  FORMAT (20X,1H$,48X,1H*)
001471          GO TO 692
001474     550  WRITE (6,750)
001500     750  FORMAT (20X,1H$,49X,1H*)
001500          GO TO 692
001503     551  WRITE (6,751)
001507     751  FORMAT (20X,1H$,50X,1H*)
001507          GO TO 692
001512     552  WRITE (6,752)
001516     752  FORMAT (20X,1H$,51X,1H*)
001516          GO TO 692
001521     553  WRITE (6,753)
001525     753  FORMAT (20X,1H$,52X,1H*)
001525          GO TO 692
001530     554  WRITE (6,754)
001534     754  FORMAT (20X,1H$,53X,1H*)
001534          GO TO 692
001537     555  WRITE (6,755)
001543     755  FORMAT (20X,1H$,54X,1H*)
001543          GO TO 692
001546     556  WRITE (6,756)
001552     756  FORMAT (20X,1H$,55X,1H*)
001552          GO TO 692
001555     557  WRITE (6,757)
001561     757  FORMAT (20X,1H$,56X,1H*)
001561          GO TO 692
001564     558  WRITE (6,758)
001570     758  FORMAT (20X,1H$,57X,1H*)
001570          GO TO 692
```

```
001577      759 FORMAT (20X,1H$,58X,1H*)
001577          GO TO 692
001602      560 WRITE (6,760)
001606      760 FORMAT (20X,1H$,59X,1H*)
001606          GO TO 692
001611      561 WRITE (6,761)
001615      761 FORMAT (20X,1H$,60X,1H*)
001615          GO TO 692
001620      562 WRITE (6,762)
001624      762 FORMAT (20X,1H$,61X,1H*)
001624          GO TO 692
001627      563 WRITE (6,763)
001633      763 FORMAT (20X,1H$,62X,1H*)
001633          GO TO 692
001636      564 WRITE (6,764)
001642      764 FORMAT (20X,1H$,63X,1H*)
001642          GO TO 692
001645      565 WRITE (6,765)
001651      765 FORMAT (20X,1H$,64X,1H*)
001651          GO TO 692
001654      566 WRITE (6,766)
001660      766 FORMAT (20X,1H$,65X,1H*)
001660          GO TO 692
001663      567 WRITE (6,767)
001657      767 FORMAT (20X,1H$,66X,1H*)
001667          GO TO 692
001672      568 WRITE (6,768)
001676      768 FORMAT (20X,1H$,67X,1H*)
001676          GO TO 692
001701      569 WRITE (6,769)
001705      769 FORMAT (20X,1H$,68X,1H*)
001705          GO TO 692
001710      570 WRITE (6,770)
001714      770 FORMAT (20X,1H$,69X,1H*)
001714          GO TO 692
001717      571 WRITE (6,771)
001723      771 FORMAT (20X,1H$,70X,1H*)
001723          GO TO 692
001726      572 WRITE (6,772)
001732      772 FORMAT (20X,1H$,71X,1H*)
001732          GO TO 692
001735      573 WRITE (6,773)
001741      773 FORMAT (20X,1H$,72X,1H*)
001741          GO TO 692
001744      574 WRITE (6,774)
001750      774 FORMAT (20X,1H$,73X,1H*)
001750          GO TO 692
001753      575 WRITE (6,775)
001757      775 FORMAT (20X,1H$,74X,1H*)
001757          GO TO 692
001762      576 WRITE (6,776)
001766      776 FORMAT (20X,1H$,75X,1H*)
001766          GO TO 692
001771      577 WRITE (6,777)
001775      777 FORMAT (20X,1H$,76X,1H*)
001775          GO TO 692
002000      578 WRITE (6,778)
```

```
002004     778 FORMAT (20X,1H$,77X,1H*)
002004         GO TO 692
002007     579 WRITE (6,779)
002013     779 FORMAT (20X,1H$,78X,1H*)
002013         GO TO 692
002016     580 WRITE (6,780)
002022     780 FORMAT (20X,1H$,79X,1H*)
002022         GO TO 692
002025     600 WRITE (6,800) XMAX2
002033     800 FORMAT (10X,F10.3,1H*)
002033         GO TO 692
002036     601 WRITE (6,801) XMAX2
002044     801 FORMAT (10X,F10.3,1H+,    1H*)
002044         GO TO 692
002047     602 WRITE (6,802) XMAX2
002055     802 FORMAT (10X,F10.3,1H+, 1X,1H*)
002055         GO TO 692
002060     603 WRITE (6,803) XMAX2
002066     803 FORMAT (10X,F10.3,1H+, 2X,1H*)
002066         GO TO 692
002071     604 WRITE (6,804) XMAX2
002077     804 FORMAT (10X,F10.3,1H+, 3X,1H*)
002077         GO TO 692
002102     605 WRITE (6,805) XMAX2
002110     805 FORMAT (10X,F10.3,1H+, 4X,1H*)
002110         GO TO 692
002113     606 WRITE (6,806) XMAX2
002121     806 FORMAT (10X,F10.3,1H+, 5X,1H*)
002121         GO TO 692
002124     607 WRITE (6,807) XMAX2
002132     807 FORMAT (10X,F10.3,1H+, 6X,1H*)
002132         GO TO 692
002135     608 WRITE (6,808) XMAX2
002143     808 FORMAT (10X,F10.3,1H+, 7X,1H*)
002143         GO TO 692
002146     609 WRITE (6,809) XMAX2
002154     809 FORMAT (10X,F10.3,1H+, 8X,1H*)
002154         GO TO 692
002157     610 WRITE (6,810) XMAX2
002165     810 FORMAT (10X,F10.3,1H+, 9X,1H*)
002165         GO TO 692
002170     611 WRITE (6,811) XMAX2
002176     811 FORMAT (10X,F10.3,1H+,10X,1H*)
002176         GO TO 692
002201     612 WRITE (6,812) XMAX2
002207     812 FORMAT (10X,F10.3,1H+,11X,1H*)
002207         GO TO 692
002212     613 WRITE (6,813) XMAX2
002220     813 FORMAT (10X,F10.3,1H+,12X,1H*)
002220         GO TO 692
002223     614 WRITE (6,814) XMAX2
002231     814 FORMAT (10X,F10.3,1H+,13X,1H*)
002231         GO TO 692
002234     615 WRITE (6,815) XMAX2
002242     815 FORMAT (10X,F10.3,1H+,14X,1H*)
002242         GO TO 692
002245     616 WRITE (6,816) XMAX2
002253     816 FORMAT (10X,F10.3,1H+,15X,1H*)
```

```
002264    817 FORMAT (10X,F10.3,1H+,16X,1H*)
002264        GO TO 692
002267    618 WRITE (6,818) XMAX2
002275    818 FORMAT (10X,F10.3,1H+,17X,1H*)
002275        GO TO 692
002300    619 WRITE (6,819) XMAX2
002306    819 FORMAT (10X,F10.3,1H+,18X,1H*)
002306        GO TO 692
002311    620 WRITE (6,820) XMAX2
002317    820 FORMAT (10X,F10.3,1H+,19X,1H*)
002317        GO TO 692
002322    621 WRITE (6,821) XMAX2
002330    821 FORMAT (10X,F10.3,1H+,20X,1H*)
002330        GO TO 692
002333    622 WRITE (6,822) XMAX2
002341    822 FORMAT (10X,F10.3,1H+,21X,1H*)
002341        GO TO 692
002344    623 WRITE (6,823) XMAX2
002352    823 FORMAT (10X,F10.3,1H+,22X,1H*)
002352        GO TO 692
002355    624 WRITE (6,824) XMAX2
002363    824 FORMAT (10X,F10.3,1H+,23X,1H*)
002363        GO TO 692
002366    625 WRITE (6,825) XMAX2
002374    825 FORMAT (10X,F10.3,1H+,24X,1H*)
002374        GO TO 692
002377    626 WRITE (6,826) XMAX2
002405    826 FORMAT (10X,F10.3,1H+,25X,1H*)
002405        GO TO 692
002410    627 WRITE (6,827) XMAX2
002416    827 FORMAT (10X,F10.3,1H+,26X,1H*)
002416        GO TO 692
002421    628 WRITE (6,828) XMAX2
002427    828 FORMAT (10X,F10.3,1H+,27X,1H*)
002427        GO TO 692
002432    629 WRITE (6,829) XMAX2
002440    829 FORMAT (10X,F10.3,1H+,28X,1H*)
002440        GO TO 692
002443    630 WRITE (6,830) XMAX2
002451    830 FORMAT (10X,F10.3,1H+,29X,1H*)
002451        GO TO 692
002454    631 WRITE (6,831) XMAX2
002462    831 FORMAT (10X,F10.3,1H+,30X,1H*)
002462        GO TO 692
002465    632 WRITE (6,832) XMAX2
002473    832 FORMAT (10X,F10.3,1H+,31X,1H*)
002473        GO TO 692
002476    633 WRITE (6,833) XMAX2
002504    833 FORMAT (10X,F10.3,1H+,32X,1H*)
002504        GO TO 692
002507    634 WRITE (6,834) XMAX2
002515    834 FORMAT (10X,F10.3,1H+,33X,1H*)
002515        GO TO 692
002520    635 WRITE (6,835) XMAX2
002526    835 FORMAT (10X,F10.3,1H+,34X,1H*)
002526        GO TO 692
```

```
002531      636 WRITE (6,836) XMAX2
002537      836 FORMAT (10X,F10.3,1H+,35X,1H*)
002537          GO TO 692
002542      637 WRITE (6,837) XMAX2
002550      837 FORMAT (10X,F10.3,1H+,36X,1H*)
002550          GO TO 692
002553      638 WRITE (6,838) XMAX2
002561      838 FORMAT (10X,F10.3,1H+,37X,1H*)
002561          GO TO 692
002564      639 WRITE (6,839) XMAX2
002572      839 FORMAT (10X,F10.3,1H+,38X,1H*)
002572          GO TO 692
002575      640 WRITE (6,840) XMAX2
002603      840 FORMAT (10X,F10.3,1H+,39X,1H*)
002603          GO TO 692
002606      641 WRITE (6,841) XMAX2
002614      841 FORMAT (10X,F10.3,1H+,40X,1H*)
002614          GO TO 692
002617      642 WRITE (6,842) XMAX2
002625      842 FORMAT (10X,F10.3,1H+,41X,1H*)
002625          GO TO 692
002630      643 WRITE (6,843) XMAX2
002636      843 FORMAT (10X,F10.3,1H+,42X,1H*)
002636          GO TO 692
002641      644 WRITE (6,844) XMAX2
002647      844 FORMAT (10X,F10.3,1H+,43X,1H*)
002647          GO TO 692
002652      645 WRITE (6,845) XMAX2
002660      845 FORMAT (10X,F10.3,1H+,44X,1H*)
002660          GO TO 692
002663      646 WRITE (6,846) XMAX2
002671      846 FORMAT (10X,F10.3,1H+,45X,1H*)
002671          GO TO 692
002674      647 WRITE (6,847) XMAX2
002702      847 FORMAT (10X,F10.3,1H+,46X,1H*)
002702          GO TO 692
002705      648 WRITE (6,848) XMAX2
002713      848 FORMAT (10X,F10.3,1H+,47X,1H*)
002713          GO TO 692
002716      649 WRITE (6,849) XMAX2
002724      849 FORMAT (10X,F10.3,1H+,48X,1H*)
002724          GO TO 692
002727      650 WRITE (6,850) XMAX2
002735      850 FORMAT (10X,F10.3,1H+,49X,1H*)
002735          GO TO 692
002740      651 WRITE (6,851) XMAX2
002746      851 FORMAT (10X,F10.3,1H+,50X,1H*)
002746          GO TO 692
002751      652 WRITE (6,852) XMAX2
002757      852 FORMAT (10X,F10.3,1H+,51X,1H*)
002757          GO TO 692
002762      653 WRITE (6,853) XMAX2
002770      853 FORMAT (10X,F10.3,1H+,52X,1H*)
002770          GO TO 692
002773      654 WRITE (6,854) XMAX2
003001      854 FORMAT (10X,F10.3,1H+,53X,1H*)
003001          GO TO 692
003004      655 WRITE (6,855) XMAX2
```

```
003012          GO TO 692
003015      656 WRITE (6,856) XMAX2
003023      856 FORMAT (10X,F10.3,1H+,55X,1H*)
003023          GO TO 692
003026      657 WRITE (6,857) XMAX2
003034      857 FORMAT (10X,F10.3,1H+,56X,1H*)
003034          GO TO 692
003037      658 WRITE (6,858) XMAX2
003045      858 FORMAT (10X,F10.3,1H+,57X,1H*)
003045          GO TO 692
003050      659 WRITE (6,859) XMAX2
003056      859 FORMAT (10X,F10.3,1H+,58X,1H*)
003056          GO TO 692
003061      660 WRITE (6,860) XMAX2
003067      860 FORMAT (10X,F10.3,1H+,59X,1H*)
003067          GO TO 692
003072      661 WRITE (6,861) XMAX2
003100      861 FORMAT (10X,F10.3,1H+,60X,1H*)
003100          GO TO 692
003103      662 WRITE (6,862) XMAX2
003111      862 FORMAT (10X,F10.3,1H+,61X,1H*)
003111          GO TO 692
003114      663 WRITE (6,863) XMAX2
003122      863 FORMAT (10X,F10.3,1H+,62X,1H*)
003122          GO TO 692
003125      664 WRITE (6,864) XMAX2
003133      864 FORMAT (10X,F10.3,1H+,63X,1H*)
003133          GO TO 692
003136      665 WRITE (6,865) XMAX2
003144      865 FORMAT (10X,F10.3,1H+,64X,1H*)
003144          GO TO 692
003147      666 WRITE (6,866) XMAX2
003155      866 FORMAT (10X,F10.3,1H+,65X,1H*)
003155          GO TO 692
003160      667 WRITE (6,867) XMAX2
003166      867 FORMAT (10X,F10.3,1H+,66X,1H*)
003166          GO TO 692
003171      668 WRITE (6,868) XMAX2
003177      868 FORMAT (10X,F10.3,1H+,67X,1H*)
003177          GO TO 692
003202      669 WRITE (6,869) XMAX2
003210      869 FORMAT (10X,F10.3,1H+,68X,1H*)
003210          GO TO 692
003213      670 WRITE (6,870) XMAX2
003221      870 FORMAT (10X,F10.3,1H+,69X,1H*)
003221          GO TO 692
003224      671 WRITE (6,871) XMAX2
003232      871 FORMAT (10X,F10.3,1H+,70X,1H*)
003232          GO TO 692
003235      672 WRITE (6,872) XMAX2
003243      872 FORMAT (10X,F10.3,1H+,71X,1H*)
003243          GO TO 692
003246      673 WRITE (6,873) XMAX2
003254      873 FORMAT (10X,F10.3,1H+,72X,1H*)
003254          GO TO 692
003257      674 WRITE (6,874) XMAX2
003265      874 FORMAT (10X,F10.3,1H+,73X,1H*)
```

```
003265         GO TO 692
003270     675 WRITE (6,875) XMAX2
003276     875 FORMAT (10X,F10.3,1H+,74X,1H*)
003276         GO TO 692
003301     676 WRITE (6,876) XMAX2
003307     876 FORMAT (10X,F10.3,1H+,75X,1H*)
003307         GO TO 692
003312     677 WRITE (6,877) XMAX2
003320     877 FORMAT (10X,F10.3,1H+,76X,1H*)
003320         GO TO 692
003323     678 WRITE (6,878) XMAX2
003331     878 FORMAT (10X,F10.3,1H+,77X,1H*)
003331         GO TO 692
003334     679 WRITE (6,879) XMAX2
003342     879 FORMAT (10X,F10.3,1H+,78X,1H*)
003342         GO TO 692
003345     680 WRITE (6,880) XMAX2
003353     880 FORMAT (10X,F10.3,1H+,79X,1H*)
003353     692 J = J+1
003355     698 CONTINUE
        C      PRINT X AXIS
003362         XMAX2 = XMAXO -(XINTO*8.)
003365     905 WRITE (6,910)  XMAX2
003373     910 FORMAT (10X,F10.3,1H+,8(10H---------+)5X,10HCONSTRAINT)
        C       FIND SCALE FOR X AXIS AND PRINT
003373         DO 920 I = 1,9
003377         AI = I
003400         SCL(I) = XMINC +(XINTC * (AI-1.))
003404     920 CONTINUE
003406         WRITE (6,930) SCL(1),SCL(2),SCL(3),SCL(4),SCL(5),SCL(6),SCL(7),
              1 SCL(8),SCL(9)
003434     930 FORMAT (16X,F8.3,8(2X,F8.3))
003434     999 RETURN
003435         END
```

```
000007         DIMENSION A(20,NX)
000007         DO 500 J = 1,NX
000010         IF (A(IX,J)) 510,500,510
000014     500 CONTINUE
000017         ISH = 0
000017         RETURN
000020     510 ISH = 1
000021         RETURN
000022         END
```

```
                 SUBROUTINE TRAJ (KINIT,KLAST,Y)
000006           COMMON TM
000006            DIMENSIONSC(50),Y(50),X(50),B(50)
000006           NOC = KLAST
000007           DO 25 K = KINIT,KLAST
000010           CK=K
000011           X(K)=CK*TM
000013        25 CONTINUE
000015           DO 50 K = KINIT,KLAST
000016        50 B(K) = Y(K)
000022           DO 60 J = 1,NOC
000024           K=NOC-J
000026           DO 60 I=KINIT,K
000027        58 IF (B(I) -B(I+1)) 60,60,59
000033        59 TEMP = B(I)
000035           B(I) = B(I+1)
000037           B(I+1) = TEMP
000041        60 CONTINUE
000046           FK=KLAST+1
000050           XMAX=FK*TM
000051           XRNG = XMAX
000053           XMIN = C.0
000053           YMAX = B(NOC)
000055           YMIN = B(KINIT)
000057           IF(YMAX-YMIN)67,68,67
000062        68 YRNG = ABS(YMIN)*2.
000065           GO TO 69
000065        67 YRNG=ABS(YMAX-YMIN)
000070        69 YINT=YRNG/6.
000072           XINT = XRNG /8.
000074           XINT6 = XINT/6.
000076           WRITE (6,70)
000101        70 FORMAT(1H1,15HTRAJECTORY PLOT)
000101           DO 73 I = 1,7
000105           AI = I
000106        73 SC(I)= YMIN+(YINT * (AI-1.))
000114           WRITE (6,75) SC(1),SC(2),SC(3),SC(4),SC(5),SC(6),SC(7)
000136        75 FORMAT (1H0, 22X,7F10.3)
000136           WRITE (6,80)
000142        80 FORMAT (29X,1H+,6(10H---------+))
000142           YINT10=YINT/10.
000144           J = KINIT
000146           K3 = 0
000150           DO 755 K=1,48
000151           K3 = K3+1
000153           XMAXU=XMIN+(XINT6*(AK-1.))
000156           AK=K
000160           XMAXL=XMIN+(XINT6*AK)
000161           IF ( J.GT.NOC) GO TO 85
000165           IF (X(J) .GE. XMAXU .AND. X(J) .LE. XMAXL) GO TO 150
000176        85 IF (K3-6) 740,90,740
000200        90 K3 = 0
000201           GO TO 750
000202       150 DO 710 I=1,7
000204           AI=I
000205           YMIN1= YMIN+(YINT * (AI-1.))
000211           YMIN2=YMIN+(YINT*AI)
```

```
000214          IF(Y(J) .GE. YMIN1.AND.Y(J).LT. YMIN2) GO TO 160
000226          GO TO 710
000226      160 DO 700 I1=1,10
000230          AI1=I1
000231          YMIN11=YMIN1+(YINT10 * (AI1 - 1.))
000235          YMIN12 = YMIN1 + (YINT10*AI1)
000240          IF (Y(J) .GE. YMIN11 .AND. Y(J) .LT. YMIN12) GO TO 180
000252          GO TO 700
000252      180 IF (K3-6) 290,190,290
000254      190 K3 = 0
000255          GO TO 295
000256      290 GO TO (301,302,303,304,305,306,560),I
000271      295 GO TO (311,312,313,314,315,316,860),I
000304      301 GO TO  (500,501,502,503,504,505,506,507,508,509) ,I1
000322      302 GO TO  (510,511,512,513,514,515,516,517,518,519) ,I1
000340      303 GO TO  (520,521,522,523,524,525,526,527,528,529) ,I1
000356      304 GO TO  (530,531,532,533,534,535,536,537,538,539) ,I1
000374      305 GO TO  (540,541,542,543,544,545,546,547,548,549) ,I1
000412      306 GO TO  (550,551,552,553,554,555,556,557,558,559) ,I1
000430      311 GO TO (800,801,802,803,804,805,806,807,808,809),I1
000446      312 GO TO (810,811,812,813,814,815,816,817,818,819),I1
000464      313 GO TO (820,821,822,823,824,825,826,827,828,829),I1
000502      314 GO TO (830,831,832,833,834,835,836,837,838,839),I1
000520      315 GO TO (840,841,842,843,844,845,846,847,848,849),I1
000536      316 GO TO (850,851,852,853,854,855,856,857,858,859),I1
000554      500 WRITE (6,600)
000560      600 FORMAT (29X,1H*,29X,1H≤)
000560          GO TO 720
000563      501 WRITE (6,601)
000567      601 FORMAT (30X,1H*,28X,1H≤)
000567          GO TO 720
000572      502 WRITE (6,602)
000576      602 FORMAT (31X,1H*,27X,1H≤)
000576          GO TO 720
000601      503 WRITE (6,603)
000605      603 FORMAT (32X,1H*,26X,1H≤)
000605          GO TO 720
000610      504 WRITE (6,604)
000614      604 FORMAT (33X,1H*,25X,1H≤)
000614          GO TO 720
000617      505 WRITE (6,605)
000623      605 FORMAT (34X,1H*,24X,1H≤)
000623          GO TO 720
000625      506 WRITE (6,606)
000632      606 FORMAT (35X,1H*,23X,1H≤)
000632          GO TO 720
000635      507 WRITE (6,607)
000641      607 FORMAT (36X,1H*,22X,1H≤)
000641          GO TO 720
000644      508 WRITE (6,608)
000650      608 FORMAT (37X,1H*,21X,1H≤)
000650          GO TO 720
000653      509 WRITE (6,609)
000657      609 FORMAT (38X,1H*,20X,1H≤)
000657          GO TO 720
000662      510 WRITE (6,610)
000666      610 FORMAT (39X,1H*,19X,1H≤)
000666          GO TO 720
```

```
000671      511 WRITE (6,611)
000675      611 FORMAT (40X,1H*,18X,1H≤)
000675          GO TO 720
000700      512 WRITE (6,612)
000704      612 FORMAT (41X,1H*,17X,1H≤)
000704          GO TO 720
000707      513 WRITE (6,613)
000713      613 FORMAT (42X,1H*,16X,1H≤)
000713          GO TO 720
000716      514 WRITE (6,614)
000722      614 FORMAT (43X,1H*,15X,1H≤)
000722          GO TO 720
000725      515 WRITE (6,615)
000731      615 FORMAT (44X,1H*,14X,1H≤)
000731          GO TO 720
000734      516 WRITE (6,616)
000740      616 FORMAT (45X,1H*,13X,1H≤)
000740          GO TO 720
000743      517 WRITE (6,617)
000747      617 FORMAT (46X,1H*,12X,1H≤)
000747          GO TO 720
000752      518 WRITE (6,618)
000756      618 FORMAT (47X,1H*,11X,1H≤)
000756          GO TO 720
000761      519 WRITE (6,619)
000765      619 FORMAT (48X,1H*,10X,1H≤)
000765          GO TO 720
000770      520 WRITE (6,620)
000774      620 FORMAT (49X,1H*, 9X,1H≤)
000774          GO TO 720
000777      521 WRITE (6,621)
001003      621 FORMAT (50X,1H*, 8X,1H≤)
001003          GO TO 720
001006      522 WRITE (6,622)
001012      622 FORMAT (51X,1H*, 7X,1H≤)
001012          GO TO 720
001015      523 WRITE (6,623)
001021      623 FORMAT (52X,1H*, 6X,1H≤)
001021          GO TO 720
001024      524 WRITE (6,624)
001030      624 FORMAT (53X,1H*, 5X,1H≤)
001030          GO TO 720
001033      525 WRITE (6,625)
001037      625 FORMAT (54X,1H*, 4X,1H≤)
001037          GO TO 720
001042      526 WRITE (6,626)
001046      626 FORMAT (55X,1H*, 3X,1H≤)
001046          GO TO 720
001051      527 WRITE (6,627)
001055      627 FORMAT (56X,1H*, 2X,1H≤)
001055          GO TO 720
001060      528 WRITE (6,628)
001064      628 FORMAT (57X,1H*, 1X,1H≤)
001064          GO TO 720
001067      529 WRITE (6,629)
001073      629 FORMAT (58X,1H*,    1H≤)
001073          GO TO 720
001076      530 WRITE (6,630)
```

```
001102            GO TO 720
001105        531 WRITE (6,631)
001111        631 FORMAT (59X,1H≤,    1H*)
001111            GO TO 720
001114        532 WRITE (6,632)
001120        632 FORMAT (59X,1H≤, 1X,1H*)
001120            GO TO 720
001123        533 WRITE (6,633)
001127        633 FORMAT (59X,1H≤, 2X,1H*)
001127            GO TO 720
001132        534 WRITE (6,634)
001136        634 FORMAT (59X,1H≤, 3X,1H*)
001136            GO TO 720
001141        535 WRITE (6,635)
001145        635 FORMAT (59X,1H≤, 4X,1H*)
001145            GO TO 720
001150        536 WRITE (6,636)
001154        636 FORMAT (59X,1H≤, 5X,1H*)
001154            GO TO 720
001157        537 WRITE (6,637)
001163        637 FORMAT (59X,1H≤, 6X,1H*)
001163            GO TO 720
001166        538 WRITE (6,638)
001172        638 FORMAT (59X,1H≤, 7X,1H*)
001172            GO TO 720
001175        539 WRITE (6,639)
001201        639 FORMAT (59X,1H≤, 8X,1H*)
001201            GO TO 720
001204        540 WRITE (6,640)
001210        640 FORMAT (59X,1H≤, 9X,1H*)
001210            GO TO 720
001213        541 WRITE (6,641)
001217        641 FORMAT (59X,1H≤,10X,1H*)
001217            GO TO 720
001222        542 WRITE (6,642)
001226        642 FORMAT (59X,1H≤,11X,1H*)
001226            GO TO 720
001231        543 WRITE (6,643)
001235        643 FORMAT (59X,1H≤,12X,1H*)
001235            GO TO 720
001240        544 WRITE (6,644)
001244        644 FORMAT (59X,1H≤,13X,1H*)
001244            GO TO 720
001247        545 WRITE (6,645)
001253        645 FORMAT (59X,1H≤,14X,1H*)
001253            GO TO 720
001256        546 WRITE (6,646)
001262        646 FORMAT (59X,1H≤,15X,1H*)
001262            GO TO 720
001265        547 WRITE (6,647)
001271        647 FORMAT (59X,1H≤,16X,1H*)
001271            GO TO 720
001274        548 WRITE (6,648)
001300        648 FORMAT (59X,1H≤,17X,1H*)
001300            GO TO 720
001303        549 WRITE (6,649)
001307        649 FORMAT (59X,1H≤,18X,1H*)
```

```
001307          GO TO 720
001312      550 WRITE (6,650)
001316      650 FORMAT (59X,1H≤,19X,1H*)
001316          GO TO 720
001321      551 WRITE (6,651)
001325      651 FORMAT (59X,1H≤,20X,1H*)
001325          GO TO 720
001330      552 WRITE (6,652)
001334      652 FORMAT (59X,1H≤,21X,1H*)
001334          GO TO 720
001337      553 WRITE (6,653)
001343      653 FORMAT (59X,1H≤,22X,1H*)
001343          GO TO 720
001346      554 WRITE (6,654)
001352      654 FORMAT (59X,1H≤,23X,1H*)
001352          GO TO 720
001355      555 WRITE (6,655)
001361      655 FORMAT (59X,1H≤,24X,1H*)
001361          GO TO 720
001364      556 WRITE (6,656)
001370      656 FORMAT (59X,1H≤,25X,1H*)
001370          GO TO 720
001373      557 WRITE (6,657)
001377      657 FORMAT (59X,1H≤,26X,1H*)
001377          GO TO 720
001402      558 WRITE (6,658)
001406      658 FORMAT (59X,1H≤,27X,1H*)
001406          GO TO 720
001411      559 WRITE (6,659)
001415      659 FORMAT (59X,1H≤,28X,1H*)
001415          GO TO 720
001420      560 WRITE (6,660 )
001424      660 FORMAT (59X,1H≤,29X,1H*)
001424          GO TO 720
001427      800 WRITE (6,900) XMAXL
001435      900 FORMAT (16X,F10.3, 3X,1H*,29X,1H+)
001435          GO TO 720
001440      801 WRITE (6,901) XMAXL
001446      901 FORMAT (16X,F10.3, 4X,1H*,28X,1H+)
001446          GO TO 720
001451      802 WRITE (6,902) XMAXL
001457      902 FORMAT (16X,F10.3, 5X,1H*,27X,1H+)
001457          GO TO 720
001462      803 WRITE (6,903) XMAXL
001470      903 FORMAT (16X,F10.3, 6X,1H*,26X,1H+)
001470          GO TO 720
001473      804 WRITE (6,904) XMAXL
001501      904 FORMAT (16X,F10.3, 7X,1H*,25X,1H+)
001501          GO TO 720
001504      805 WRITE (6,905) XMAXL
001512      905 FORMAT (16X,F10.3, 8X,1H*,24X,1H+)
001512          GO TO 720
001515      806 WRITE (6,906) XMAXL
001523      906 FORMAT (16X,F10.3, 9X,1H*,23X,1H+)
001523          GO TO 720
001526      807 WRITE (6,907) XMAXL
001534      907 FORMAT (16X,F10.3,10X,1H*,22X,1H+)
001534          GO TO 720
```

```
001545            GO TO 720
001550        809 WRITE (6,909) XMAXL
001556        909 FORMAT (16X,F10.3,12X,1H*,20X,1H+)
001556            GO TO 720
001561        810 WRITE (6,910) XMAXL
001567        910 FORMAT (16X,F10.3,13X,1H*,19X,1H+)
001567            GO TO 720
001572        811 WRITE (6,911) XMAXL
001600        911 FORMAT (16X,F10.3,14X,1H*,18X,1H+)
001600            GO TO 720
001603        812 WRITE (6,912) XMAXL
001611        912 FORMAT (16X,F10.3,15X,1H*,17X,1H+)
001611            GO TO 720
001614        813 WRITE (6,913) XMAXL
001622        913 FORMAT (16X,F10.3,16X,1H*,16X,1H+)
001622            GO TO 720
001625        814 WRITE (6,914) XMAXL
001633        914 FORMAT (16X,F10.3,17X,1H*,15X,1H+)
001633            GO TO 720
001636        815 WRITE (6,915) XMAXL
001644        915 FORMAT (16X,F10.3,18X,1H*,14X,1H+)
001644            GO TO 720
001647        816 WRITE (6,916) XMAXL
001655        916 FORMAT (16X,F10.3,19X,1H*,13X,1H+)
001655            GO TO 720
001660        817 WRITE (6,917) XMAXL
001666        917 FORMAT (16X,F10.3,20X,1H*,12X,1H+)
001666            GO TO 720
001671        818 WRITE (6,918) XMAXL
001677        918 FORMAT (16X,F10.3,21X,1H*,11X,1H+)
001677            GO TO 720
001702        819 WRITE (6,919) XMAXL
001710        919 FORMAT (16X,F10.3,22X,1H*,10X,1H+)
001710            GO TO 720
001713        820 WRITE (6,920) XMAXL
001721        920 FORMAT (16X,F10.3,23X,1H*, 9X,1H+)
001721            GO TO 720
001724        821 WRITE (6,921) XMAXL
001732        921 FORMAT (16X,F10.3,24X,1H*, 8X,1H+)
001732            GO TO 720
001735        822 WRITE (6,922) XMAXL
001743        922 FORMAT (16X,F10.3,25X,1H*, 7X,1H+)
001743            GO TO 720
001746        823 WRITE (6,923) XMAXL
001754        923 FORMAT (16X,F10.3,26X,1H*, 6X,1H+)
001754            GO TO 720
001757        824 WRITE (6,924) XMAXL
001765        924 FORMAT (16X,F10.3,27X,1H*, 5X,1H+)
001765            GO TO 720
001770        825 WRITE (6,925) XMAXL
001776        925 FORMAT (16X,F10.3,28X,1H*, 4X,1H+)
001776            GO TO 720
002001        826 WRITE (6,926) XMAXL
002007        926 FORMAT (16X,F10.3,29X,1H*, 3X,1H+)
002007            GO TO 720
002012        827 WRITE (6,927) XMAXL
```

```
002020     927 FORMAT (16X,F10.3,30X,1H*, 2X,1H+)
002020         GO TO 720
002023     828 WRITE (6,928) XMAXL
002031     928 FORMAT (16X,F10.3,31X,1H*, 1X,1H+)
002031         GO TO 720
002034     829 WRITE (6,929) XMAXL
002042     929 FORMAT (16X,F10.3,32X,1H*,    1H+)
002042         GO TO 720
002045     830 WRITE (6,930) XMAXL
002053     930 FORMAT(16X,F10.3,33X,1H*)
002053         GO TO 720
002056     831 WRITE (6,931) XMAXL
002064     931 FORMAT (16X,F10.3,33X,1H+,    1H*)
002064         GO TO 720
002067     832 WRITE (6,932) XMAXL
002075     932 FORMAT (16X,F10.3,33X,1H+, 1X,1H*)
002075         GO TO 720
002100     833 WRITE (6,933) XMAXL
002106     933 FORMAT (16X,F10.3,33X,1H+, 2X,1H*)
002106         GO TO 720
002111     834 WRITE (6,934) XMAXL
002117     934 FORMAT (16X,F10.3,33X,1H+, 3X,1H*)
002117         GO TO 720
002122     835 WRITE (6,935) XMAXL
002130     935 FORMAT (16X,F10.3,33X,1H+, 4X,1H*)
002130         GO TO 720
002133     836 WRITE (6,936) XMAXL
002141     936 FORMAT (16X,F10.3,33X,1H+, 5X,1H*)
002141         GO TO 720
002144     837 WRITE (6,937) XMAXL
002152     937 FORMAT (16X,F10.3,33X,1H+, 6X,1H*)
002152         GO TO 720
002155     838 WRITE (6,938) XMAXL
002163     938 FORMAT (16X,F10.3,33X,1H+, 7X,1H*)
002163         GO TO 720
002166     839 WRITE (6,939) XMAXL
002174     939 FORMAT (16X,F10.3,33X,1H+, 8X,1H*)
002174         GO TO 720
002177     840 WRITE (6,940) XMAXL
002205     940 FORMAT (16X,F10.3,33X,1H+, 9X,1H*)
002205         GO TO 720
002210     841 WRITE (6,941) XMAXL
002216     941 FORMAT (16X,F10.3,33X,1H+,10X,1H*)
002216         GO TO 720
002221     842 WRITE (6,942) XMAXL
002227     942 FORMAT (16X,F10.3,33X,1H+,11X,1H*)
002227         GO TO 720
002232     843 WRITE (6,943) XMAXL
002240     943 FORMAT (16X,F10.3,33X,1H+,12X,1H*)
002240         GO TO 720
002243     844 WRITE (6,944) XMAXL
002251     944 FORMAT (16X,F10.3,33X,1H+,13X,1H*)
002251         GO TO 720
002254     845 WRITE (6,945) XMAXL
002262     945 FORMAT (16X,F10.3,33X,1H+,14X,1H*)
002262         GO TO 720
002265     846 WRITE (6,946) XMAXL
002273     946 FORMAT (16X,F10.3,33X,1H+,15X,1H*)
```

```
002276      847 WRITE (6,947) XMAXL
002304      947 FORMAT (16X,F10.3,33X,1H+,16X,1H*)
002304          GO TO 720
002307      848 WRITE (6,948) XMAXL
002315      948 FORMAT (16X,F10.3,33X,1H+,17X,1H*)
002315          GO TO 720
002320      849 WRITE (6,949) XMAXL
002326      949 FORMAT (16X,F10.3,33X,1H+,18X,1H*)
002326          GO TO 720
002331      850 WRITE (6,950) XMAXL
002337      950 FORMAT (16X,F10.3,33X,1H+,19X,1H*)
002337          GO TO 720
002342      851 WRITE (6,951) XMAXL
002350      951 FORMAT (16X,F10.3,33X,1H+,20X,1H*)
002350          GO TO 720
002353      852 WRITE (6,952) XMAXL
002361      952 FORMAT (16X,F10.3,33X,1H+,21X,1H*)
002361          GO TO 720
002364      853 WRITE (6,953) XMAXL
002372      953 FORMAT (16X,F10.3,33X,1H+,22X,1H*)
002372          GO TO 720
002375      854 WRITE (6,954) XMAXL
002403      954 FORMAT (16X,F10.3,33X,1H+,23X,1H*)
002403          GO TO 720
002406      855 WRITE (6,955) XMAXL
002414      955 FORMAT (16X,F10.3,33X,1H+,24X,1H*)
002414          GO TO 720
002417      856 WRITE (6,956) XMAXL
002425      956 FORMAT (16X,F10.3,33X,1H+,25X,1H*)
002425          GO TO 720
002430      857 WRITE (6,957) XMAXL
002436      957 FORMAT (16X,F10.3,33X,1H+,26X,1H*)
002436          GO TO 720
002441      858 WRITE (6,958) XMAXL
002447      958 FORMAT (16X,F10.3,33X,1H+,27X,1H*)
002447          GO TO 720
002452      859 WRITE (6,959) XMAXL
002460      959 FORMAT (16X,F10.3,33X,1H+,28X,1H*)
002460      860 WRITE (6,960) XMAXL
002466      960 FORMAT (16X,F10.3,33X,1H+,29X,1H*)
002466          GO TO 720
002471      700 CONTINUE
002473      710 CONTINUE
002475      720 J= J+1
002477          GO TO 755
002477      740 WRITE (6,742)
002503      742 FORMAT (59X,1HS)
002503          GO TO 755
002506      750 WRITE (6,752) XMAXL
002514      752 FORMAT (16X,F10.3,33X,1H+)
002514      755 CONTINUE
002520          WRITE (6,760)
002524      760 FORMAT(58X,4HTIME)
002524      999 RETURN
002525          END
```


S. GOVERNMENT PRINTING OFFICE: 1973-739-028/46

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546